OTIMIZAÇÃO PRÁTICA USANDO O LAGRANGIANO AUMENTADO

José Mario Martínez

Departamento de Matemática Aplicada
IMECC-UNICAMP

22 de novembro de 2006. Atualizado: 11 de julho de 2007; 12 de agosto de 2008, 21 de outubro de 2008, 31 de março de 2009, 14 de abril de 2009, 3 de agosto de 2009.

Prefácio

Estas notas se destinam a fornecer os fundamentos teóricos de ALGENCAN, o método de Lagrangiano Aumentado do Projeto TANGO. (Veja

`www.ime.usp.br/~egbirgin/tango`

ou

`www.ime.unicamp.br/~martinez/software`.)

Ao mesmo tempo, procuram colocar na mesma perspectiva as Condições de Otimalidade de Programação Não-Linear e os critérios de parada usados em algoritmos práticos.

A pergunta subjacente em cada capítulo destas notas é: De acordo com a Teoria, o quê acontece na Prática?

A Teoria é totalmente desenvolvida no mundo do contínuo, essencialmente $\mathbb{R}^n$, enquanto a Prática o faz no universo paralelo dos números representáveis em ponto flutuante. Portanto, há fenônenos da Prática que a Teoria não pode prever. Nosso ponto de vista é explorar ao máximo a explicação dos comportamentos práticos que *sim* podem ser explicados pela Teoria. Consequentemente, consideramos que a Teoria relevante é a que contribui a explicar fenômenos práticos. Diante de cada resultado teórico nos perguntaremos: De acordo com isto, quê deve acontecer com programas computacionais que implementam este algoritmo? Se a resposta for pouco significativa, o resultado teórico será irrelevante ou incompleto.

Na análise do Método de Lagrangiano Aumentado seguimos, implicitamente, o seguinte *Protocolo*, o qual pode ser aplicado à análise teórica de qualquer algoritmo de Programação Não Linear:

(a) Estudar em qué condições o algoritmo encontra minimizadores globais. Alguns algoritmos, como o de Lagrangiano Aumentado, convergem a minimizadores globais se se supõe que minimizadores globais de problemas mais simples podem ser encontrados. Acreditamos que ésta é uma virtude teórica com implicações práticas.

(b) Estudar as propriedades de convergência do algoritmo do ponto de vista da *admissibilidade*. A pergunta é se o método converge a pontos que satisfazem as restrições do problema. Deve haver uma resposta mesmo que o problema não possua pontos viáveis, por exemplo, o algoritmo pode converger a pontos estacionários de uma medida de inviabilidade. Se se supõe limitação da seqüência gerada pelo algoritmo, deve haver condições suficientes razoáveis relativas ao problema que garantam que a limitação da seqüência acontece.

(c) Provar que se um ponto limite viável satisfaz uma condição de qualificação, as condições KKT são satisfeitas nesse ponto. A condição de qualificação deve ser tão fraca quanto possível. Devem ser identificadas condições satisfeitas pela seqüência gerada pelo algoritmo que possam ser usadas como critério de parada.

(d) Provar que se o algoritmo converge a um ponto estacionário com condições adicionais tão fracas quanto possível, boas propriedades de estabilidade ou velocidade são satisfeitas.

O problema de Programação Não-Linear a que nos referimos neste texto é o de minimizar uma função contínua com restrições de igualdade e desigualdade, contínuas. Em nenhum momento faremos suposições de convexidade. O Método de Lagrangiano Aumentado possui uma teoria extremamente rica quando aplicado a problemas convexos (ver, por exemplo, o survey de Iusem [18]) mas ela não será considerada aqui.

O Método de Lagrangiano Aumentado não é um método de moda na pesquisa corrente (ano 2006) de Programação Não-Linear. Entretanto, as modas passam, e este método possui um conjunto de características difíceis de emular por outros algoritmos de PNL:

1. Alto grau de modularidade: a eficiência do método depende em grande medida da eficiência de algoritmos para minimizar em caixas. O problema de minimizar em caixas é relativamente simples e os progressos nesta área são permanentes. Tais progressos se refletem rapidamente quando incorporados à metodologia de Lagrangiano Aumentado.

2. Uma conseqüência do ítem anterior que merece destaque se refere aos problemas de *grande porte*, onde o número de variáveis ou o número de restrições é muito grande. Neste caso, a dificuldade se translada totalmente ao subproblema de minimizar com restrições simples. Para este subproblema é usual que bons algoritmos para lidar com o grande porte possam ser definidos.

3. O método pode ser usado quando as restrições são naturalmente divididas em dois grupos: fáceis e difíceis. Neste caso, as restrições fáceis permanecem em um *nível inferior* e a técnica de Lagrangiano Aumentado se aplica às restrições difíceis. Os subproblemas podem ser resolvidos por qualquer método sabidamente eficiente com esse tipo de restrições simples. O caso de caixas nada mais é que um caso particular de restrição simples.

4. O método de Lagrangiano Aumentado pode ser usado sem derivadas com preservação de teoria rigorosa de convergência.

5. A convergência a minimizadores globais depende apenas de que essa propriedade esteja presente para os subproblemas simples.

6. Características que dificultam a aplicação de outros métodos (como o excesso de restrições com a consequente diminuição de graus de liberdade) não afetam o método de Lagrangiano Aumentado.

7. O método pode ser usado com plena teoria de convergência para alguns problemas não diferenciáveis de importância prática, como o de Otimização do Menor Valor Ordenado.

Evidentemente, a reputação de um método de Optimização está ligada à qualidade dos programas que o implementam. Por esse motivo, nosso software `Algencan`, do projeto TANGO, é objeto de atualização permanente, muitas vezes orientada pelas inquietudes e dificuldades dos usuários. Muitos problemas práticos de Programação Não-Linear não são fáceis e sua abordagem ingênua costuma redundar em fracasso. Entretanto, muitas vezes o sucesso vem de pequenas manipulações que, para o usuário experimentado, acabam sendo familiares.

As notas estão em elaboração, e sua atualização depende de sucessivos seminários realizados em Campinas, Bahía Blanca, Novi Sad, Foz de Iguaçu. Provisoriamente, devo registrar meus agradecimentos a Viviana Ramírez, Gabriel Haeser, Shetty, Laura Schuverdt, Lucas Pedroso, Juliano Francisco, Ernesto Birgin, Roberto Andreani, Sergio Ventura, Sandra Santos, Rodrigo Lima, Kleysson Andreotti, Marcia A. Gomes-Ruggiero, Ana Friedlander, Leandro Prudente, Jair Silva, Margarida Pinheiro Mello, Wesley Wakabayashi, Luiz Antonio Medeiros.

Sumário

Capítulo 1

Introdução

O problema de Programação Não-Linear (PNL) consiste em

$$\text{Minimizar } f(x)$$

sujeita a $h_i(x) = 0, i = 1, \ldots, m,\ g_i(x) \leq 0, i = 1, \ldots, p,\ x \in \Omega$, onde Ω é um subconjunto de $I\!R^n$, geralmente simples. Muitas vezes Ω é o próprio $I\!R^n$ e, freqüentemente, Ω é um paralelepípedo n-dimensional (ou caixa). As funções f, h_i, g_i são contínuas e, em geral, deriváveis.

Este problema tem muitas aplicações práticas e existem muitos métodos para resolvê-lo. O Método de Lagrangiano Aumentado é uma versão moderna de uma das idéias mais antigas sobre como lidar com ele. Tal idéia consiste em eliminar as restrições $h_i(x) = 0$ e $g_i(x) \leq 0$, incluíndo estas restrições na função objetivo de maneira que o problema assim transformado tenha soluções iguais ou parecidas às do problema original. Dado um parâmetro "de penalidade" $\rho > 0$, a função que denominamos Lagrangiano Aumentado se define assim:

$$L_\rho(x, \lambda, \mu) = f(x) + \frac{\rho}{2}\left\{ \sum_{i=1}^{m} \left[h_i(x) + \frac{\lambda_i}{\rho} \right]^2 + \sum_{i=1}^{p} \left[\max\left(0, g_i(x) + \frac{\mu_i}{\rho}\right) \right]^2 \right\}.$$

Os parâmetros $\lambda_i \in I\!R$ e $\mu_i \geq 0$ se chamam "multiplicadores". Aqui definimos L com um único parâmetro de penalidade ρ, embora, às vezes, prefira-se definir um penalizador diferente ρ_i para cada restrição. Quando se inicia a execução do Método de Lagrangiano Aumentado, os multiplicadores iniciais e o penalizador são dados, de maneira que os passos fundamentais do método são:

1. Minimizar $L_\rho(x, \lambda, \mu)$ sujeita a $x \in \Omega$.

2. Decidir se o ponto obtido no primeiro passo pode ser aceito como solução do problema original. Em caso afirmativo, parar a execução do algoritmo.

3. De acordo com os resultados dos passos anteriores, atualizar os multiplicadores e o parâmetro de penalidade.

4. Voltar ao primeiro passo.

É fácil perceber que, pelo menos no caso em que os multiplicadores são nulos, a estratégia é muito sensata.

Os métodos de Lagrangiano Aumentado seguem, quase sempre, o esquema acima. Devem ser diferenciados de outros métodos para PNL que usam o Lagrangiano Aumentado como função de mérito auxiliar.

As definições básicas usadas ao longo deste texto serão as de minimizador global e local.

Diremos que x^* é minimizador global de $f(x)$ sujeita a $x \in D$ quando $x^* \in D$ e $f(x) \geq f(x^*)$ para todo $x \in D$.

O ponto $x^* \in D$ será minimizador local desse problema quando existe $\varepsilon > 0$ tal que $f(x) \geq f(x^*)$ sempre que $x \in D$ e a distância entre x e x^* é menor que ε.

Notações:

$\| \cdot \|$ será sempre a norma Euclidiana. Muitas vezes poderá ser substituída por uma norma arbitrária.

As componentes de um vetor serão denotadas com subíndices.

Se $v \in I\!R^n$, denotaremos v_+ o vetor cujas componentes são $\max\{0, v_1\}, \dots, \max\{0, v_n\}$.

$I\!R$ é o conjunto dos números reais, $I\!N$ é o conjunto dos naturais, começando com 0.

$K \underset{\infty}{\subset} I\!N$ indica que K é um subconjunto infinito de $I\!N$, preservando a ordem. Rigorosamente falando, K é uma seqüência estritamente crescente de números naturais. Analogamente, $K_1 \underset{\infty}{\subset} K$, etc.

Capítulo 2

Método de Penalidade Externa

Neste capítulo consideramos a versão mais básica do chamado Método de Penalidade Externa. Apesar desta estratégia raramente poder ser usada na prática, suas características teóricas iluminam algoritmos mais sofisticados, inclusive o Método de Lagrangiano Aumentado, nosso objeto de estudo. Veremos também que as propriedades do Método de Penalidade Externa fornecem provas simples de resultados bastante fortes de otimalidade.

Seja $\Omega \subset I\!R^n$ um conjunto fechado. Sejam $f : \Omega \to I\!R$, $h : \Omega \to I\!R^m$, $g : \Omega \to I\!R^p$.

Escreveremos $h = (h_1, \ldots, h_m)^T$, $g = (g_1, \ldots, g_p)^T$ e suporemos que as funções f, h, g são contínuas em Ω.

Definimos o problema PNL da seguinte maneira:

$$\text{Minimizar } f(x)$$

$$\text{sujeita a } h(x) = 0,\ g(x) \leq 0, x \in \Omega.$$

O problema PNL é o problema central encarado neste livro.

2.1 Penalidade Externa Pura

O Método de Penalidade Externa está definido pelo Algoritmo 2.1.

Algoritmo 2.1

Seja $\{\rho_k\}$ uma seqüência de números positivos tal que

$$\lim_{k \to \infty} \rho_k = \infty.$$

Para todo $k \in I\!N$, o iterando x^k é uma solução global do *subproblema*

$$\text{Minimizar } f(x) + \frac{\rho_k}{2}[\|h(x)\|^2 + \|g(x)_+\|^2] \text{ sujeita a } x \in \Omega. \tag{2.1}$$

Observe que, na descrição do Método de Penalidade Externa dada no Algoritmo 2.1, não existe um "critério de parada". Isto significa que, teoricamente, a seqüência $\{x^k\}$ tem infinitos termos, onde não se exclui que muitos deles sejam iguais. Implementações práticas, por outro lado, devem estar equipadas com critérios de parada, como veremos em capítulos posteriores.

No Método de Penalidade Externa substituímos o problema original por uma seqüência de problemas nos quais o conjunto de restrições foi reduzido a

$$x \in \Omega.$$

Em implementações práticas, o conjunto Ω será, geralmente, "simples", de maneira que cada subproblema definido pelo Método de Penalidade Externa seja mais fácil que o problema original PNL. Entretanto, do ponto de vista teórico, não há restrições sobre a complexidade de Ω.

Observamos que, formalmente, o iterando x^{k+1} do método não está conectado em absoluto com o iterando x^k. Na prática, entretanto, esta conexão sempre existirá, pois x^k é o "ponto inicial" natural para resolver o subproblema de maneira iterativa.

A exigência de obtenção de um minimizador global do subproblema é um enorme empecilho para a implementação deste método. Na prática, salvo casos muito específicos, isto é muito difícil. Mesmo do ponto de vista teórico, a existência de uma solução global do subproblema somente é garantida supondo, por exemplo, a compacidade do conjunto simples Ω. Quando x^k existe para todo $k \in \mathbb{N}$ dizemos que o método está *bem definido*.

A idéia da penalidade externa é substituír a função objetivo original f por uma função onde *se castiga* a não-admissibilidade de x. Quanto menos admissível for o ponto x com respeito às restrições $h(x) = 0$ e $g(x) \leq 0$, mais castigado ele será, mediante um aumento da função objetivo do subproblema. A penalidade é maior quanto maior for ρ_k. Podemos interpretar que, quando ρ_k é muito grande, a função objetivo do subproblema coincide com $f(x)$ se x é admissível e se aproxima de infinito se x não é admissível. A dificuldade dos subproblemas aumenta junto com o valor de ρ_k. Apesar de que, em casos simples, começar com valores enormes de ρ_k dá bons resultados, em situações mais complexas valores extremos de ρ_k fazem com que o termo $f(x)$ no subproblema seja "cancelado" pelo termo de penalidade, pelo menos quando se trabalha com aproximações no computador.

Independentemente disso, o que nos interessa neste capítulo são as propriedades teóricas do método tal como definido acima. Existem duas perguntas básicas que, de fato, são comuns a todos os algoritmos de otimização com restrições:

- O método encontra (converge a) pontos admissíveis?
- O método encontra (converge a) pontos ótimos?

A primeira resposta não pode ser sempre positiva porque, às vezes, o problema original não tem pontos admissíveis. Logo, nosso interesse seria que o método encontrasse pontos admissíveis quando estes existem e que, em caso contrário, encontrasse algo que nos sirva para garantir a não-existência de tais pontos. Os teoremas 2.1 e 2.2 mostram que, felizmente, isso é o que acontece com o Método de Penalidade Externa.

Nos Teoremas 2.1–2.5 suporemos que a seqüência $\{x^k\}$ está bem definida pelo Algoritmo 2.1.

Teorema 2.1. *Suponhamos que a seqüência* $\{x^k\}$ *admite um ponto limite* x^*. *Então,* x^* *é uma solução global do problema*

$$\textit{Minimizar } \|h(x)\|^2 + \|g(x)_+\|^2 \textit{ sujeita a } x \in \Omega. \tag{2.2}$$

Prova. Seja $K \underset{\infty}{\subset} I\!N$ tal que

$$\lim_{k \in K} x^k = x^*.$$

Como $x^k \to x^*$ para $k \in K$, $x^k \in \Omega$ para todo k, e Ω é fechado , temos que $x^* \in \Omega$.

Suponhamos que x^* não seja um minimizador global de (2.2). Isto implica que existe $z \in \Omega$ tal que

$$\|h(x^*)\|^2 + \|g(x^*)_+\|^2 > \|h(z)\|^2 + \|g(z)_+\|^2.$$

Como h e g são contínuas, existe $c > 0$ tal que, para $k \in K$ suficientemente grande,

$$\|h(x^k)\|^2 + \|g(x^k)_+\|^2 > \|h(z)\|^2 + \|g(z)_+\|^2 + c.$$

Portanto, para esses índices k,

$$f(x^k)+\frac{\rho_k}{2}[\|h(x^k)\|^2+\|g(x^k)_+\|^2] > f(z)+\frac{\rho_k}{2}[\|h(z)\|^2+\|g(z)_+\|^2]+\frac{\rho_k c}{2}+f(x^k)-f(z).$$

Agora, como $\{x^k\}_{k \in K}$ é um conjunto limitado, f é contínua e $\rho_k \to \infty$, temos que, para $k \in K$ suficientemente grande,

$$\frac{\rho_k c}{2} + f(x^k) - f(z) > 0.$$

Logo,

$$f(x^k) + \frac{\rho_k}{2}[\|h(x^k)\|^2 + \|g(x^k)_+\|^2] > f(z) + \frac{\rho_k}{2}[\|h(z)\|^2 + \|g(z)_+\|^2],$$

o que contradiz o fato de que x^k é uma solução global do subproblema definido por ρ_k. Esta contradição decorre de supor que x^* não era um minimizador global de (2.2). **QED**

Teorema 2.2. *Suponhamos que a região admissível do problema PNL é não-vazia. Suponhamos, além disso, que a seqüência $\{x^k\}$ admite um ponto limite x^*. Então, x^* é admissível.*

Prova. Pelo Teorema 2.1, x^* é minimizador global de (2.2). Mas, como existe pelo menos um ponto admissível z, neste ponto a função objetivo de (2.2) vale zero. Portanto, essa função também deve se anular em x^*, o que implica que x^* é admissível. **QED**

Pelo Teorema 2.2, se existem pontos admissíveis e a seqüência gerada pelo algoritmo é limitada, um dos pontos admissíveis será fatalmente encontrado pelo algoritmo. Por outro lado, o Teorema 2.1 garante que, se a região admissível for vazia, um ponto que minimiza a soma de quadrados das inadmissibilidades (ou seja, o "menos inadmissível") será achado .

No Teorema 2.4 veremos que, quando a região admissível é não-vazia, o Algoritmo 2.1 encontra um minimizador global do problema PNL. Este resultado será conseqüência de uma propriedade mais forte, dada no Teorema 2.3. Nele veremos que os pontos limite do Método de Penalidade Externa minimizam $f(x)$ no conjunto dos pontos menos inadmissíveis.

Teorema 2.3. *Seja $z \in \Omega$ um minimizador global de (2.2). Suponhamos que a seqüência $\{x^k\}$ admite um ponto limite x^*. Então x^* também é um minimizador global de (2.2) e, além disso, $f(x^*) \leq f(z)$.*

Prova. O fato de que x^* é minimizador global de (2.2) foi provado no Teorema 2.1.

Seja $K \underset{\infty}{\subset} I\!N$ tal que $\lim_{k \in K} x^k = x^*$.

Pela definição de x^k, temos que:

$$f(x^k) + \frac{\rho_k}{2}[\|h(x^k)\|^2 + \|g(x^k)_+\|^2] \leq f(z) + \frac{\rho_k}{2}[\|h(z)\|^2 + \|g(z)_+\|^2],$$

para todo $k \in K$.

Mas, como z é um minimizador global de (2.2),

$$\frac{\rho_k}{2}[\|h(x^k)\|^2 + \|g(x^k)_+\|^2] \geq \frac{\rho_k}{2}[\|h(z)\|^2 + \|g(z)_+\|^2].$$

Portanto,

$$f(x^k) \leq f(z) \; \forall \; k \in K.$$

Tomando limites nesta desigualdade, pela continuidade de f e o fato de que $x^k \to x^*$ para $k \in K$, obtemos que $f(x^*) \leq f(z)$. **QED**

Teorema 2.4. *Suponhamos que a região admissível do problema PNL é não-vazia e a seqüência $\{x^k\}$ admite um ponto limite x^*. Então, x^* é um minimizador global de PNL.*

Prova. Pelo Teorema 2.3, x^* é admissível e $f(x^*) \leq f(z)$ para qualquer outro ponto admissível $z \in \Omega$. Portanto, x^* é minimizador global de PNL. **QED**

Usualmente, os iterandos x^k do Método de Penalidade Externa não são pontos admissíveis. Ou seja, alguma restrição do tipo $g_i(x^k) \leq 0$ ou $h_i(x^k) = 0$ quase sempre é violada. Quando isto não é assim, ou seja, quando todas as restrições são satisfeitas em um iterando x^k, algo bastante surpreendente acontece: x^k deve ser solução do problema PNL. Provamos esta propriedade no Teorema 2.5.

Teorema 2.5. *Suponhamos que existe $k \in I\!N$ tal que x^k é admissível. Então, x^k é minimizador global de PNL.*

Prova. Seja $z \in \Omega$ um ponto admissível. Pela definição de x^k temos:

$$f(x^k) + \frac{\rho_k}{2}[\|h(x^k)\|^2 + \|g(x^k)_+\|^2] \leq f(z) + \frac{\rho_k}{2}[\|h(z)\|^2 + \|g(z)_+\|^2].$$

Mas, como x^k e z são admissíveis,

$$\|h(x^k)\|^2 + \|g(x^k)_+\|^2 = \|h(z)\|^2 + \|g(z)_+\|^2 = 0.$$

Portanto:

$$f(x^k) \leq f(z).$$

Como z é um ponto admissível arbitrário, deduzimos que x^k é um minimizador global de PNL. **QED**

2.2 Penalidade Externa com Controle de Admissibilidade

Dado que trabalhar com parâmetros de penalidade muito grandes é, via de regra, inconveniente, procuraremos aumentar o parâmetro de penalidade só se for estritamente necessário. Julgaremos que aumentar ρ_k é desnecessário se a medida de inadmissibilidade de x^k é suficientemente menor que a medida de inadmissibilidade de x^{k-1}.

Algoritmo 2.2.

Seja $x^0 \in \Omega$ um ponto inicial arbitrário, $\rho_1 > 0$ um parâmetro inicial de penalidade, $\tau \in [0, 1)$ e $\gamma > 1$. Inicializar o "contador" $k \leftarrow 1$.

Passo 1. Encontrar $x^k \in \Omega$ solução global do subproblema (2.1).

Passo 2. Se

$$\max\{\|h(x^k)\|_\infty, \|g(x^k)_+\|_\infty\} \leq \tau \max\{\|h(x^{k-1})\|_\infty, \|g(x^{k-1})_+\|_\infty\},$$

definir $\rho_{k+1} = \rho_k$. Em caso contrário, definir $\rho_{k+1} = \gamma\rho_k$.

Passo 3. Atualizar $k \leftarrow k + 1$ e voltar ao Passo 1.

No teste do Passo 2 não é obrigatório usar $\|\cdot\|_\infty$, embora esta norma seja a mais freqüente nas aplicações práticas.

O leitor cético pode argumentar que não há nenhuma diferença essencial entre o Algoritmo 2.2 e o Algoritmo 2.1. De fato, se em determinada iteração o parâmetro de penalidade não é aumentado, o subproblema na iteração seguinte será o mesmo e, portanto, o mesmo ponto pode ser escolhido como solução. No novo teste, a menos que x^k seja admissível (e, portanto, ótimo) a redução da inadmissibilidade não poderá ocorrer (pois $x^k = x^{k-1}$) e, em conseqüência, o parâmetro de penalidade será aumentado. Assim, a seqüência ρ_k tenderá a infinito e estaremos inteiramente sob as hipóteses do Algoritmo 2.1.

Entretanto, de acordo com o Algoritmo 2.2, se o teste de melhora de admissibilidade for bem sucedido, o ponto x^{k+1} *pode* mas *não precisa* ser escolhido igual a x^k. Ou seja, o subproblema (2.1) embora sendo o mesmo nas duas iterações, pode ter mais de uma solução global e nada impede que na iteração $k+1$ seja escolhida uma solução diferente daquela escolhida na iteração k. Consideremos o exemplo em uma variável definido por:

$$f(x) = -x^2, \; m = 1, p = 0, \; h_1(x) = x, \Omega = I\!R.$$

Começando com $\rho_1 = 2$ e $x^0 = 100$, a função objetivo do subproblema é nula. Logo podemos escolher o minimizador global $x^1 = 10$. O teste de melhora de admissibilidade dá resultado positivo e, portanto, $\rho_2 = \rho_1 = 2$. Como a função objetivo do subproblema continua sendo a função nula, podemos escolher agora a solução $x^k = 1$ quando $k = 2$. De novo, o teste é positivo e assim por diante. Neste caso, a seqüência gerada pelo Algoritmo 2.2 poderia ser $\{100, 10, 1, 0.1, 0.01, 0.001, \ldots\}$ e o parâmetro de penalidade não aumentaria nunca.

Claro que não é a existência de exemplos mais ou menos bizarros o que nos motiva a introduzir este algoritmo. Sua presença aqui se justifica, como veremos, em função da futura definição de outros métodos onde o teste de admissibilidade desempenha um papel mais importante.

Se, no Algoritmo 2.2, o parâmetro de penalidade precisa ser aumentado infinitas vezes, o algoritmo se reduz ao Algoritmo 2.1. Portanto, suas propriedades, dadas pelos Teoremas 2.1–2.5, são as mesmas. Logo, a única análise adicional que merece o novo algoritmo corresponde ao caso em que, a partir de certo k_0, o parâmetro de penalidade não aumenta mais. O Teorema 2.6 explica o que acontece nesse caso.

Teorema 2.6. *Suponhamos que a seqüência $\{x^k\}$ está bem definida pelo Algoritmo 2.2 e que, para todo $k \geq k_0$, temos que $\rho_k = \rho_{k_0}$. Suponhamos que x^* é um ponto limite da seqüência. Então, x^* é um minimizador global de PNL.*

Prova. Para todo $k \geq k_0$ o teste de melhora da admissibilidade deve ter dado resultado positivo, portanto:

$$\max\{\|h(x^k)\|_\infty, \|g(x^k)_+\|_\infty\} \leq \tau \ \max\{\|h(x^{k-1})\|_\infty, \|g(x^{k-1})_+\|_\infty\}$$

para todo $k \geq k_0$. Portanto, como $\tau < 1$,

$$\lim_{k\to\infty} \max\{\|h(x^k)\|_\infty, \|g(x^k)_+\|_\infty\} = 0.$$

Logo, se $\lim_{k\in K} x^k = x^*$,

$$\lim_{k\in K} \max\{\|h(x^k)\|_\infty, \|g(x^k)_+\|_\infty\} = 0.$$

Pela continuidade de h e g, isto implica que $\|h(x^*)\| = \|g(x^*)_+\| = 0$. Ou seja, x^* é admissível.

Seja $z \in \Omega$ outro ponto admissível arbitrário. Pela definição do algoritmo,

$$\begin{aligned} f(x^k) \leq f(x^k) + \rho_{k_0}[\|h(x^k)\|^2 + \|g(x^k)_+\|^2] &= f(x^k) + \rho_k[\|h(x^k)\|^2 + \|g(x^k)_+\|^2] \\ &\leq f(z) + \rho_k[\|h(z)\|^2 + \|g(z)_+\|^2] = f(z). \end{aligned}$$

Logo, $f(x^k) \leq f(z)$ para todo $k \geq k_0$. Tomando limites nesta desigualdade, pela continuidade de f, temos que $f(x^*) \leq f(z)$. Como z era um ponto admissível arbitrário, se deduz que x^* é solução global de PNL. **QED**

EXERCÍCIOS

1. Leia os resultados exibidos neste capítulo. O que pode falhar? Procure traduzir cada resultado em termos de funcionamento prático do método analisado.

2. Considere, nos resultados deste capítulo, a possibilidade de substituír as normas $\| \cdot \|$ e $\| \cdot \|_\infty$ por normas arbitrárias. Existe algum impedimento para proceder assim?

3. Escreva um programa que implemente, de maneira tão fiel quanto for possível, os métodos de penalidade externa. Procure, com ajuda desse programa, encontrar exemplos de sucesso e fracasso do método. Escreva uma monografia explicando as formas que o fracasso toma na prática, e seu correlato teórico.

4. Considere o problema "de dois níveis"
$$\text{Minimizar } f(x)$$
sujeita à condição de que x é minimizador global de $f_1(x)$. Analise sua possível resolução através de subproblemas do tipo
$$\text{Minimizar } f(x) + \rho f_1(x).$$
Sugestão: suponha que c é um limitante inferior de $f_1(x)$ e considere a restrição $h_1(x) = 0$, onde $h_1(x) = \sqrt{f_1(x) - c}$.

5. Considere o problema de dois níveis "com restrições no nível inferior":
$$\text{Minimizar }_{x,u} f(x, u)$$
sujeita a que x é minimizador (com respeito à primeira variável) de $f_1(x, u)$ com determinadas restrições. Usando o exercício anterior, defina um método de penalidade externa dupla para resolver este problema e analise suas propriedades.

6. Acesse a página do projeto TANGO `www.ime.usp.br/~egbirgin/tango`, copie o programa `Algencan`, estude os comentários e o próprio programa tanto quanto for possível e use o programa para "verificar" os resultados teóricos enunciados neste capítulo, implementando exemplos adequados de sua invenção.

7. Prove que, no Método de Penalidade Externa, $f(x^{k+1}) \geq f(x^k)$ para todo k, que os valores das funções objetivo dos subproblemas também crescem monotonamente mas que a medida de não-admissibilidade decresce de uma iteração para outra.

Capítulo 3

Método de Penalidade Deslocada

Consideremos o problema PNL, definido no Capítulo 2 com as mesmas hipóteses de continuidade sobre f, h e g enunciadas nesse capítulo. Vimos no Teorema 2.5 que, se um iterando x^k for admissível, então esse iterando é a solução de PNL. Infelizmente, o caso em que x^k é admissível é pouco freqüente. A razão dessa falta de freqüência é fácil de entender: a função que "castiga" a não-admissibilidade de um ponto x é, no Método de Penalidade Externa, $\frac{\rho_k}{2}[\|h(x)\|^2+\|g(x)_+\|^2]$. O parâmetro de penalidade ρ_k pode ser grande, mas as medidas de não-admissibilidade $\|h(x)\|$ e $\|g(x)_+\|$ aparecem elevadas ao quadrado nesta função. Isto significa que, para pontos x levemente não-admissíveis, o castigo é brando ($\|h(x)\| < 1 \Rightarrow \|h(x)\|^2 \ll 1$ e $\|g(x)_+\| < 1 \Rightarrow \|g(x)_+\|^2 \ll 1$). Por outro lado, é provável que a função objetivo $f(x)$ tenha, no ponto não-admissível x, um valor fortemente menor que nos pontos admissíveis mais próximos (se não fosse assim, as restrições não seriam necessárias!). Em tais circunstâncias, para qualquer valor finito de ρ_k o baixo valor de $f(x)$ compensará o castigo pela não-admissibilidade. Logo, é possível afirmar que, quase sempre, o Método de Penalidade Externa aproxima a solução do problema PNL por meio de uma seqüência de pontos não-admissíveis. O leitor ficará mais convencido desta propriedade computando os iterandos em um exemplo numérico simples: Minimizar x, sujeita a $-x \leq 0$.

É um pouco frustrante o fato de que todos os iterandos de um método sejam não-admissíveis, embora o método convirja a uma solução. Isto porque, na prática, sempre paramos a execução de um algoritmo em uma iteração k, não no infinito. Em muitos problemas reais, pontos admissíveis levemente não-ótimos são toleráveis mas pontos não-admissíveis não têm utilidade alguma. Por isso, procuramos corrigir o Método de Penalidade Externa para aumentar as chances de que um iterando x^k seja solução ou, pelo menos, para que x^k seja admissível. Para fixar idéias, suponhamos que $m = 0$, $p = 1$. (Recomendamos acompanhar esta leitura com seu desenho favorito.) Se $f(x)$ desce fortemente quando passamos do ponto ótimo a pontos não-admissíveis o castigo quadrático não será suficiente para fazer admissível nenhum dos iterandos x^k. Entretanto, isto acontece porque no Método de Penalidade Externa estamos castigando o desvio de $g(x)_+$ com respeito ao vetor 0. Se, em vez disso, aplicássemos uma penalidade, não ao desvio em relação ao 0, mas ao desvio em relação a um vetor negativo w, o termo de penalidade correspondente seria $\|(g(x) - w)_+\|^2$ e o minimizador do subproblema ficaria levemente deslocado e mais perto da admissibilidade. Assim, as chances de que x^k fosse admissível e, talvez, quase ótimo, aumentariam. Fazendo o mesmo raciocínio em relação a $h(x)$ parece recomendável que a função de castigo básica, em vez de $\|h(x)\|^2 + \|g(x)_+\|^2$, seja

$$\|h(x) - v^k\|^2 + \|(g(x) - w^k)_+\|^2$$

para vetores $v^k \in I\!R^m$ e $w^k \in I\!R^p_-$ que, por enquanto, não sabemos como calcular.

No caso $v^k = 0, w^k = 0$ teremos o Método de Penalidade Externa definido no Capítulo 2. Agora, quando ρ_k é muito grande a solução do subproblema no Método de Penalidade Externa aproxima muito bem a solução do PNL. Nesse caso, não parece conveniente afastar-se demasiado de $v = 0, w = 0$. Este requisito pode ser forçado definindo:

$$v^k = -\frac{\bar{\lambda}^k}{\rho_k},\; w^k = -\frac{\bar{\mu}^k}{\rho_k}$$

e impondo a condição de que $\bar{\lambda}^k \in I\!R^m$ e $\bar{\mu}^k \in I\!R^p_+$ estejam em um conjunto limitado, independentemente de k.

Com isto, e apesar de que não sabemos calcular $\bar{\lambda}^k$ nem $\bar{\mu}^k$, podemos definir o Método de Penalidade Deslocada da seguinte maneira:

Algoritmo 3.1.

Seja $x^0 \in \Omega$ um ponto inicial arbitrário. Sejam $\lambda_{min} \leq \lambda_{max}$ e $\mu_{max} > 0$ os parâmetros que definem a limitação de $\bar{\lambda}^k$ e $\bar{\mu}^k$. Suponhamos que

$$\bar{\lambda}^1_i \in [\lambda_{min}, \lambda_{max}] \;\forall\, i = 1, \ldots, m$$

e

$$\bar{\mu}^1_i \in [0, \mu_{max}] \;\forall\, i = 1, \ldots, p.$$

Seja $\{\rho_k\}$ uma seqüência de números positivos que tende a infinito. Inicializar o "contador" $k \leftarrow 1$.

Passo 1. Encontrar $x^k \in \Omega$ solução global do subproblema

$$\text{Minimizar } f(x) + \frac{\rho_k}{2}\left[\left\|h(x) + \frac{\bar{\lambda}^k}{\rho_k}\right\|^2 + \left\|\left(g(x) + \frac{\bar{\mu}^k}{\rho_k}\right)_+\right\|^2\right] \text{ sujeita a } x \in \Omega. \quad (3.1)$$

Passo 2. Calcular $\bar{\lambda}_i^{k+1} \in [\lambda_{min}, \lambda_{max}]$ para todo $i = 1, \ldots, m$ e $\bar{\mu}_i^{k+1} \in [0, \mu_{max}]$ para todo $i = 1, \ldots, p$. Atualizar $k \leftarrow k + 1$ e voltar ao Passo 1.

Observe que, apesar de que para definir o Algoritmo 3.1 usamos o modelo do Algoritmo 2.2, não ousamos, por enquanto, reproduzir o Passo 2 daquele algoritmo. Deixaremos a possibilidade de manter sem modificação o parâmetro de penalidade para mais adiante e, por enquanto, nos contentaremos em estudar as propriedades teóricas do Algoritmo 3.1. Estas propriedades serão generalizações dos teoremas provados no Capítulo 2, o que não é surpreendente pois, de fato, o Algoritmo 2.1 é o caso particular do Algoritmo 3.1 em que $\bar{\lambda}^k = 0, \bar{\mu}^k = 0$ para todo k.

Comecemos pelos teoremas relativos à admissibilidade dos pontos limites de seqüências geradas pelo Algoritmo 3.1.

Nos Teoremas 3.1–3.5 suporemos que a seqüência $\{x^k\}$ está bem definida pelo Algoritmo 3.1.

Teorema 3.1. *Suponhamos que a seqüência $\{x^k\}$ admite um ponto limite x^*. Então, x^* é uma solução global do problema*

$$\text{Minimizar } \|h(x)\|^2 + \|g(x)_+\|^2 \text{ sujeita a } x \in \Omega. \tag{3.2}$$

Prova. Esta prova generaliza a do Teorema 2.1. Seja $K \underset{\infty}{\subset} I\!N$ tal que

$$\lim_{k \in K} x^k = x^*.$$

Como $x^k \to x^*$ para $k \in K$, $x^k \in \Omega$ para todo k, e Ω é fechado , temos que $x^* \in \Omega$.

Suponhamos que x^* não é um minimizador global de (3.2). Portanto, existe $z \in \Omega$ tal que

$$\|h(x^*)\|^2 + \|g(x^*)_+\|^2 > \|h(z)\|^2 + \|g(z)_+\|^2.$$

Como h e g são contínuas e, além disso, $\bar{\lambda}^k$ e $\bar{\mu}^k$ são limitadas e $\rho_k \to \infty$, existe $c > 0$ tal que para $k \in K$ suficientemente grande:

$$\left\|h(x^k) + \frac{\bar{\lambda}^k}{\rho_k}\right\|^2 + \left\|\left(g(x^k) + \frac{\bar{\mu}^k}{\rho_k}\right)_+\right\|^2 > \left\|h(z) + \frac{\bar{\lambda}^k}{\rho_k}\right\|^2 + \left\|\left(g(z) + \frac{\bar{\mu}^k}{\rho_k}\right)_+\right\|^2 + c.$$

Portanto, para esses índices k,

$$f(x^k) + \frac{\rho_k}{2}\left[\left\|h(x^k) + \frac{\bar{\lambda}^k}{\rho_k}\right\|^2 + \left\|\left(g(x^k) + \frac{\bar{\mu}^k}{\rho_k}\right)_+\right\|^2\right]$$

$$> f(z) + \frac{\rho_k}{2}\left[\left\|h(z) + \frac{\bar{\lambda}^k}{\rho_k}\right\|^2 + \left\|\left(g(z) + \frac{\bar{\mu}^k}{\rho_k}\right)_+\right\|^2\right] + \frac{\rho_k c}{2} + f(x^k) - f(z).$$

Agora, como $\lim_{k\in K} x^k = x^*$ e f é contínua, temos que, para $k \in K$ suficientemente grande:

$$\frac{\rho_k c}{2} + f(x^k) - f(z) > 0.$$

Logo,

$$f(x^k)+\frac{\rho_k}{2}\left[\left\|h(x^k)+\frac{\bar{\lambda}^k}{\rho_k}\right\|^2+\left\|\left(g(x^k)+\frac{\bar{\mu}^k}{\rho_k}\right)_+\right\|^2\right] > f(z)+\frac{\rho_k}{2}\left[\left\|h(z)+\frac{\bar{\lambda}^k}{\rho_k}\right\|^2+\left\|\left(g(z)+\frac{\bar{\mu}^k}{\rho_k}\right)_+\right\|^2\right],$$

o que contradiz o fato de que x^k é uma solução global do subproblema definido por ρ_k. Esta contradição veio de supor que x^* podia não ser um minimizador global de (3.2). **QED**

No seguinte teorema mostramos que o Algoritmo 3.1 encontra pontos admissíveis, quando estes existem.

Teorema 3.2. *Suponhamos que a região admissível do problema PNL é não-vazia. Suponhamos, além disso, que a seqüência $\{x^k\}$ admite um ponto limite x^*. Então, x^* é admissível.*

Prova. Idêntica à prova do Teorema 2.2. **QED**

Nos agradaria provar que, como no Teorema 2.3, os pontos limites do Algoritmo 3.1 minimizam a função objetivo no conjunto dos minimizadores de (3.2). Entretanto, esta propriedade não é verdade. Considere, por exemplo, o problema PNL definido por $n = 1$, $m = 3$, $p = 0$, $f(x) = -x$, $h_1(x) = x - 1$, $h_2(x) = x + 1$, $h_3(x) = 2(x^2 - 1)$, $\Omega = I\!R$. A região admissível é obviamente vazia e os minimizadores de (3.2) são dois pontos: $-\sqrt{3}/2$ e $\sqrt{3}/2$. Portanto, o minimizador da função objetivo restrito ao conjunto dos minimizadores da inadmissibilidade é o ponto $x^* = \sqrt{3}/2$. Entretanto, se aplicarmos o Algoritmo 3.1 com $\bar{\lambda}_1^k = 2$, $\bar{\lambda}_2^k = \bar{\lambda}_3^k = 0$ para todo k, veremos que a seqüência $\{x^k\}$ converge a $-\sqrt{3}/\sqrt{2}$. Com efeito, neste caso,

$$\begin{aligned} L_{\rho_k}(x, \bar{\lambda}^k, \bar{\mu}^k) &= f(x) + h_1(x)\bar{\lambda}_1^k + \frac{\rho_k}{2}\|h(x)\|^2 + c \\ &= -x + 2(x-1) + \frac{\rho_k}{2}\|h(x)\|^2 + c, \end{aligned}$$

onde c é uma constante. A função $\|h(x)\|^2$ é par neste caso, portanto toma o mesmo valor em pontos x próximos a $\sqrt{3}/2$ que nos correspondentes pontos $-x$, próximos a $-\sqrt{3}/2$. Por isso, para cada x próximo a $\sqrt{3}/2$ existe um ponto próximo a $-\sqrt{3}/2$ onde L_{ρ_k} vale menos. Desse modo, o único ponto de acumulação possível é $-\sqrt{3}/2$.

Felizmente, a impossibilidade de generalizar o Teorema 2.3 não é tão relevante. O realmente importante é que o algoritmo tenha a propriedade de minimizar f no caso em que a região admissível é não-vazia. A propriedade de minimizar f no conjunto dos minimizadores de (3.2) pode ser sacrificada se o algoritmo tiver outras virtudes (o que ainda não foi comprovado!)

Vejamos que, com efeito, os pontos limites do Algoritmo 3.1 são soluções de PNL no caso em que há pontos admissíveis. Portanto, apesar de que a tese do Teorema 2.3 não é válida, a generalização do Teorema 2.4 se sustenta plenamente.

Teorema 3.3. *Suponhamos que a região admissível do problema PNL é não-vazia e a seqüência $\{x^k\}$ admite um ponto limite x^*. Então, x^* é um minimizador global de PNL.*

Prova. Seja $K \underset{\infty}{\subset} I\!N$ tal que $\lim_{k \in K} x^k = x^*$. Pelo Teorema 3.2, x^* é admissível. Seja $z \in \Omega$ outro ponto admissível arbitrário. Pela definição do algoritmo, temos que

$$f(x^k) + \frac{\rho_k}{2}\left[\left\|h(x^k) + \frac{\bar{\lambda}^k}{\rho_k}\right\|^2 + \left\|\left(g(x^k) + \frac{\bar{\mu}^k}{\rho_k}\right)_+\right\|^2\right]$$

$$\le f(z) + \frac{\rho_k}{2}\left[\left\|h(z) + \frac{\bar{\lambda}^k}{\rho_k}\right\|^2 + \left\|\left(g(z) + \frac{\bar{\mu}^k}{\rho_k}\right)_+\right\|^2\right].$$

Mas $h(z) = 0$ e $g(z)_+ = 0$, portanto

$$\left\|h(z) + \frac{\bar{\lambda}^k}{\rho_k}\right\|^2 = \left\|\frac{\bar{\lambda}^k}{\rho_k}\right\|^2$$

e

$$\left\|\left(g(z) + \frac{\bar{\mu}^k}{\rho_k}\right)_+\right\|^2 \le \left\|\frac{\bar{\mu}^k}{\rho_k}\right\|^2.$$

Portanto,

$$f(x^k) \le f(x^k) + \frac{\rho_k}{2}\left[\left\|h(x^k) + \frac{\bar{\lambda}^k}{\rho_k}\right\|^2 + \left\|\left(g(x^k) + \frac{\bar{\mu}^k}{\rho_k}\right)_+\right\|^2\right]$$

$$\le f(z) + \frac{\|\bar{\lambda}^k\|^2}{2\rho_k} + \frac{\|\bar{\mu}^k\|^2}{2\rho_k}.$$

Tomando limites para $k \in K$ e usando que $\rho_k \to \infty$ e que $\bar{\lambda}^k$ e $\bar{\mu}^k$ são limitados, obtemos:

$$f(x^*) \le f(z).$$

Como z é um ponto admissível arbitrário, resulta que x^* é solução global de PNL, como queriamos provar. **QED**

No Capítulo 2 vimos que, no Método de Penalidade Externa, um iterando admissível necessariamente deveria ser ótimo. Essa propriedade não se mantém no Algoritmo 3.1, como vemos no seguinte exemplo. Seja $n = 1$, $m = 0$, $p = 1$, $f(x) = -x$, $g_1(x) = x$, $\Omega = I\!R$. Claramente, a única solução global no PNL assim definido é $x^* = 0$. Entretanto, se $\bar{\mu}_1^k = \rho_k = 20$. O subproblema que definirá o iterando x^k será:

$$\text{Minimizar } \; -x + 10 \max\{0, x+1\}^2,$$

cuja solução, $x^k = -1 + 1/20$, é um ponto admissível mas não ótimo.

3.1 Penalidade Deslocada com Controle de Admissibilidade

No Capítulo 2 estudamos uma modificação do método básico de Penalidade Externa que consistia em aplicar o seguinte ditado: "Se a admissibilidade melhorou suficientemente em x^k em relação a x^{k-1}, não aumente o parâmetro de penalidade". Discutamos se é possível fazer "a mesma" modificação no Algoritmo de Penalidade Deslocada. A idéia seria fazer sempre $\rho_{k+1} = \gamma \rho_k$, exceto nas iterações em que

$$\max\{\|h(x^k)\|_\infty, \|g(x^k)_+\|_\infty\} \leq \tau \max\{\|h(x^{k-1})\|_\infty, \|g(x^{k-1})_+\|_\infty\},$$

nas quais faríamos $\rho_{k+1} = \rho_k$.

No caso em que $\rho_k \to \infty$, todos os pontos limite deste algoritmo modificado seriam soluções, pois neste caso se aplicaria o Teorema 3.3. Mas vejamos que aconteceria se, a partir de certo k_0, o parâmetro de penalidade permanecesse inalterado. Pela definição do algoritmo, se $z \in \Omega$ fosse um ponto admissível arbitrário, teríamos:

$$\begin{aligned} & f(x^k) + \frac{\rho_{k_0}}{2}\left[\left\|h(x^k) + \frac{\bar{\lambda}^k}{\rho_{k_0}}\right\|^2 + \left\|\left(g(x^k) + \frac{\bar{\mu}^k}{\rho_{k_0}}\right)_+\right\|^2\right] \\ & \leq f(z) + \frac{\rho_{k_0}}{2}\left[\left\|h(z) + \frac{\bar{\lambda}^k}{\rho_{k_0}}\right\|^2 + \left\|\left(g(z) + \frac{\bar{\mu}^k}{\rho_{k_0}}\right)_+\right\|^2\right] \end{aligned}$$

para todo $k \geq k_0$. Como $g(z) \leq 0$ e $\bar{\mu}^k/\rho_{k_0} \geq 0$,

$$\left\|\left(g(z) + \frac{\bar{\mu}^k}{\rho_{k_0}}\right)_+\right\|^2 \leq \left\|\frac{\bar{\mu}^k}{\rho_{k_0}}\right\|^2,$$

portanto, dado que $h(z) = 0$,

$$f(x^k) + \frac{\rho_{k_0}}{2}\left[\left\|h(x^k) + \frac{\bar{\lambda}^k}{\rho_{k_0}}\right\|^2 + \left\|\left(g(x^k) + \frac{\bar{\mu}^k}{\rho_{k_0}}\right)_+\right\|^2\right] \leq f(z) + \frac{\rho_{k_0}}{2}\left[\left\|\frac{\bar{\lambda}^k}{\rho_{k_0}}\right\|^2 + \left\|\frac{\bar{\mu}^k}{\rho_{k_0}}\right\|^2\right]$$

para todo $k \geq k_0$. Se x^* é um ponto limite, a limitação do parâmetro de penalidade garantiria que é admissível, pois $\max\{h(x^k), g(x^k)_+\} \leq \tau \max\{h(x^{k-1}), g(x^{k-1})_+\}$ para todo k. Logo, passando ao limite na desigualdade acima, e usando uma subseqüência convergente adequada de $\bar{\lambda}^k$ e $\bar{\mu}^k$:

$$f(x^*) + \frac{\rho_{k_0}}{2}\left[\left\|\frac{\bar{\lambda}^*}{\rho_{k_0}}\right\|^2 + \left\|\left(g(x^*) + \frac{\bar{\mu}^*}{\rho_{k_0}}\right)_+\right\|^2\right] \leq f(z) + \frac{\rho_{k_0}}{2}\left[\left\|\frac{\bar{\lambda}^*}{\rho_{k_0}}\right\|^2 + \left\|\frac{\bar{\mu}^*}{\rho_{k_0}}\right\|^2\right].$$

Portanto,

$$f(x^*) + \frac{\rho_{k_0}}{2}\left\|\left(g(x^*) + \frac{\bar{\mu}^*}{\rho_{k_0}}\right)_+\right\|^2 \leq f(z) + \frac{\rho_{k_0}}{2}\left\|\frac{\bar{\mu}^*}{\rho_{k_0}}\right\|^2.$$

Infelizmente, não podemos avançar mais. Da desigualdade acima não se infere que $f(x^*) \leq f(z)$ porque, se $g_i(x^*) < 0$ e $\bar{\mu}_i^*/\rho_{k_0} > 0$, teremos que

$$(g_i(x^*) + \bar{\mu}_i^*/\rho_{k_0})_+ < \bar{\mu}_i^*/\rho_{k_0},$$

o que impossibilita tirar conclusões sobre a relação entre $f(x^*)$ e $f(z)$. Isto significa que, para que o algoritmo modificado conserve as propriedades de convergência a minimizadores globais de PNL, deve-se garantir que $\bar{\mu}_i^*$ seja zero sempre que $g_i(x^*) < 0$. Esta propriedade se chama *complementaridade* e vemos que é essencial para garantir que pontos limite sejam soluções. Tal necessidade induz a seguinte definição para o Algoritmo de Penalidade Deslocada Modificado.

Algoritmo 3.2

Seja $x^0 \in \Omega$ um ponto inicial arbitrário. Sejam $\lambda_{min} < \lambda_{max}$ e $\mu_{max} > 0$ os parâmetros que definem a limitação de $\bar{\lambda}^k$ e $\bar{\mu}^k$. Suponhamos que

$$\bar{\lambda}_i^1 \in [\lambda_{min}, \lambda_{max}] \; \forall \, i = 1, \ldots, m$$

e

$$\bar{\mu}_i^1 \in [0, \mu_{max}] \; \forall \, i = 1, \ldots, p.$$

Seja $\rho_1 > 0$. Definir $V^0 = g(x^0)_+$. Inicializar $k \leftarrow 1$.

Passo 1. Encontrar $x^k \in \Omega$ solução global do subproblema (3.1).

Passo 2. Definir

$$V_i^k = \max\left\{g_i(x^k), -\frac{\bar{\mu}_i^k}{\rho_k}\right\}, i = 1, \ldots, p.$$

Se

$$\max\{\|h(x^k)\|_\infty, \|V^k\|_\infty\} \leq \tau \; \max\{\|h(x^{k-1})\|_\infty, \|V^{k-1}\|_\infty\},$$

definir $\rho_{k+1} = \rho_k$. Em caso contrário, definir $\rho_{k+1} = \gamma\rho_k$.

Passo 3. Calcular $\bar{\lambda}_i^{k+1} \in [\lambda_{min}, \lambda_{max}]$ para todo $i = 1, \ldots, m$ e $\bar{\mu}_i^{k+1} \in [0, \mu_{max}]$ para todo $i = 1, \ldots, p$. Atualizar $k \leftarrow k + 1$ e voltar ao Passo 1.

Observe que V_i^k é uma medida conjunta da admissibilidade e complementaridade de x^k em relação à restrição $g_i(x) \leq 0$. Com efeito, V_i^k vale zero somente se $g_i(x^k) \leq 0$ e, ao mesmo tempo, $\bar{\mu}_i^k = 0$ em caso de que $g_i(x^k) < 0$.

Teorema 3.4. *Suponhamos que a região admissível do problema PNL é não-vazia e que a seqüência $\{x^k\}$ está bem definida pelo Algoritmo 3.2 e admite um ponto limite x^*. Então, x^* é um minimizador global de PNL.*

Prova. Se $\lim_{k\to\infty} \rho_k = \infty$, o algoritmo é um caso particular do Algoritmo 3.1. Portanto, a tese segue do Teorema 3.3.

Se ρ_k não tende a infinito, então, a partir de certo k_0, temos que $\rho_k = \rho_{k_0}$. Logo, pelo Passo 2 do algoritmo, $\lim_{k\to\infty} \|V^k\|_\infty = 0$ e $\lim_{k\to\infty} \|h(x^k)\|_\infty = 0$. Isto implica que

$$\lim_{k\to\infty} \|h(x^k)\|_\infty = \lim_{k\to\infty} \|g(x^k)_+\|_\infty = 0.$$

Portanto, pela continuidade de h e g, todo ponto limite deve ser admissível.

Seja $K \underset{\infty}{\subset} I\!N$ tal que $\lim_{k\in K} x^k = x^*$. Sem perda de generalidade, suponhamos que $k \geq k_0$ para todo $k \in K$. Como as seqüências $\{\bar{\lambda}^k\}$ e $\{\bar{\mu}^k\}$ estão em um compacto, existem $K_1 \underset{\infty}{\subset} K$, $\lambda^* \in [\lambda_{min}, \lambda_{max}]^m$, $\bar{\mu}^* \in [0, \mu_{max}]^p$ tais que

$$\lim_{k\in K_1} x^k = x^*, \lim_{k\in K_1} \bar{\lambda}^k = \lambda^*, \lim_{k\in K_1} \bar{\mu}^k = \bar{\mu}^*.$$

Mas, pelo Passo 2 do algoritmo, $\lim_{k\in\infty} \max\{g_i(x^k), -\bar{\mu}_i^k/\rho_{k_0}\} = 0$. Portanto, se $g_i(x^*) < 0$ devemos ter, forçosamente, que $\bar{\mu}_i^* = 0$. Em outras palavras,

$$g_i(x^*)\bar{\mu}_i^* = 0 \text{ para todo } i = 1, \ldots, p.$$

Agora, o raciocínio (frustrado) que precede o enunciado deste teorema pode ser integralmente reproduzido para chegar a:

$$f(x^*) + \frac{\rho_{k_0}}{2}\left\|\left(g(x^*) + \frac{\bar{\mu}^*}{\rho_{k_0}}\right)_+\right\|^2 \leq f(z) + \frac{\rho_{k_0}}{2}\left\|\frac{\bar{\mu}^*}{\rho_{k_0}}\right\|^2.$$

Ou seja,

$$f(x^*) + \frac{\rho_{k_0}}{2}\sum_{i=1}^{p}\left(g_i(x^*) + \frac{\bar{\mu}_i^*}{\rho_{k_0}}\right)_+^2 \leq f(z) + \frac{\rho_{k_0}}{2}\sum_{i=1}^{p}\left(\frac{\bar{\mu}_i^*}{\rho_{k_0}}\right)^2. \tag{3.3}$$

Mas

$$\left(g_i(x^*) + \frac{\bar{\mu}_i^*}{\rho_{k_0}}\right)_+^2 = \max\left\{0, g_i(x^*) + \frac{\bar{\mu}_i^*}{\rho_{k_0}}\right\}^2.$$

Portanto, se $g_i(x^*) = 0$,

$$\left(g_i(x^*) + \frac{\bar{\mu}_i^*}{\rho_{k_0}}\right)_+^2 = \left(\frac{\bar{\mu}_i^*}{\rho_{k_0}}\right)^2$$

e, se $g_i(x^*) < 0$, dado que $\bar{\mu}_i^* = 0$,

$$\left(g_i(x^*) + \frac{\bar{\mu}_i^*}{\rho_{k_0}}\right)_+^2 = 0.$$

Logo, por (3.3),

$$f(x^*) + \frac{\rho_{k_0}}{2}\left\|\left(\frac{\bar{\mu}^*}{\rho_{k_0}}\right)_+\right\|^2 = f(x^*) + \frac{\rho_{k_0}}{2}\left\|\left(g(x^*) + \frac{\bar{\mu}^*}{\rho_{k_0}}\right)_+\right\|^2 \le f(z) + \frac{\rho_{k_0}}{2}\sum_{i=1}^{p}\left(\frac{\bar{\mu}_i^*}{\rho_{k_0}}\right)^2.$$

Isto implica que $f(x^*) \le f(z)$, o que completa a prova. **QED**

3.2 Cálculo dos Deslocamentos

Até agora, a escolha dos deslocamentos v^{k+1} e w^{k+1} no Passo 2 dos Algoritmos 3.1 e 3.2 foi quase arbitrária. Escolhemos $v^{k+1} = -\bar{\lambda}^{k+1}/\rho_{k+1}$ e $w^{k+1} = -\bar{\mu}^{k+1}/\rho_{k+1}$, com $\bar{\lambda}^{k+1} \in [\lambda_{min}, \lambda_{max}]$ e $\bar{\mu}^{k+1} \in [0, \mu_{max}]$ e isso é suficiente para provar os resultados de convergência deste capítulo. Entretanto, a eficiência dos algoritmos exigirá escolhas razoáveis dos vetores $\bar{\lambda}^{k+1}$ e $\bar{\mu}^{k+1}$.

Lembremos que a motivação para introduzir deslocamentos na penalidade era que, no Método de Penalidade Externa, as soluções dos subproblemas eram quase sempre não-admissíveis. Ao penalizar em relação a um ponto levemente deslocado pareciam aumentar as chances de que a solução do subproblema fosse admissível.

Suponhamos que, ao resolver o subproblema na k-ésima iteração, obtemos x^k e que $h_i(x^k) \neq 0$. (Recomendamos que o leitor acompanhe este argumento com um desenho adequado e que, para simplificar, pense em $h_i(x^k) > 0$.) Este ponto foi obtido usando o deslocamento v_i^k. Ou seja, na função Lagrangiano Aumentado apareceu o termo $\rho_k(h_i(x) - v_i^k)^2$. Evidentemente, teriamos apreciado que $h_i(x^k)$ fosse igual a zero. Entretanto, depois do acontecido, pensamos: "Isto ocorreu porque penalizar $h_i(x)$ em relação a v_i^k era inadequado. Se tivéssemos penalizado em relação a $v_i^k - h_i(x^k)$ o resultado seria mais razoável".

Ou seja, postulamos que

$$v \text{ correto} = v_i^k - h_i(x^k).$$

Agora, o deslocamento v "correto" pode ser expressado como

$$v \text{ correto} = \frac{-\lambda \text{ correto}}{\rho_k}.$$

Como $v_i^k = -\bar{\lambda}_i^k/\rho_k$, resulta que

$$\lambda \text{ correto} = \bar{\lambda}_i^k + \rho_k h_i(x^k).$$

Não temos mais chances de usar o λ "correto" na iteração que passou, mas podemos tentar usá-lo na próxima. Por isso, definimos:

$$\lambda_i^{k+1} = \bar{\lambda}_i^k + \rho_k h_i(x^k).$$

Dado que λ_i^{k+1} não está, necessariamente, entre os limitantes λ_{min} e λ_{max}, complementamos esta definição com:

$$\bar{\lambda}_i^{k+1} = \min\{\lambda_{max}, \max\{\lambda_{min}, \lambda_i^{k+1}\}\}.$$

Vejamos agora o caso de restrições de desigualdade. Suponhamos que x^k é a solução do subproblema k. Esta solução foi obtida usando o deslocamento $w_i^k \leq 0$ para cada restrição de desigualdade. Lembremos que $w_i^k = -\bar{\mu}_i^k/\rho_k$.

Consideremos primeiro o caso em que $g_i(x^k) < w_i^k$. Assim, x^k foi obtido como minimizador de uma função que penalizava apenas o fato de que $g_i(x)$ fosse maior que w_i^k e, com este deslocamento, $g_i(x^k)$ resultou estritamente menor que w_i^k. Isto significa que não havia, na realidade, motivos para deslocar a penalidade em relação a sua posição natural. Deslocamos sem necessidade! Portanto, o próximo deslocamento deve ser nulo.

Agora consideremos o caso em que $g_i(x^k) > 0$. Ou seja, x^k é um ponto inadmissível devido à não-satisfação da restrição $g_i(x) \leq 0$. Como no caso de restrições de igualdade, isto nos leva a definir um w "correto" como:

$$w \ \text{ correto} = w_i^k - g_i(x^k).$$

Portanto, seguindo o mesmo argumento das igualdades:

$$\mu_i^{k+1} = \bar{\mu}_i^k + \rho_k g_i(x^k)$$

neste caso.

Por último, consideremos o caso $0 > g_i(x^k) > w_i^k$. Aqui o ponto é admissível em relação à restrição $g_i(x) \leq 0$, mas ele foi obtido penalizando em relação a um desvio w_i^k e g_i não atingiu o valor desse desvio. Portanto, há bons motivos para pensar que, neste caso, o deslocamento w_i^k foi exagerado e que $g_i(x^k)$ deveria ter sido igual a zero. Isto nos conduz a postular, mais uma vez, que

$$w \ \text{ correto} = w_i^k - g_i(x^k).$$

Consequentemente,

$$\mu_i^{k+1} = \bar{\mu}_i^k + \rho_k g_i(x^k)$$

Juntando os três casos, chegamos a que o novo deslocamento deve ser o deslocamento antigo menos $g_i(x^k)$ se $g_i(x^k)$ é maior ou igual que o deslocamento antigo, e deve ser igual a zero em caso contrário. Em termos do multipicador, isto é:

$$\mu_i^{k+1} = \max\{0, \bar{\mu}_i^k + \rho_k g_i(x^k)\}.$$

Finalmente,

$$\bar{\mu}_i^{k+1} = \min\{\mu_{max}, \max\{0, \mu_i^{k+1}\}\}.$$

Com estas fórmulas para definir a atualização de $\bar{\lambda}$ e $\bar{\mu}$, estamos em condições de definir um novo algoritmo de penalidade deslocada. A esta altura, podemos dizer que esta é nossa primeira versão do Método de Lagrangiano Aumentado e podemos chamar a $\bar{\mu}^k$ e $\bar{\lambda}^k$ de *Multiplicadores de Lagrange aproximados.*

Algoritmo 3.3

Seja $x^0 \in \Omega$ um ponto inicial arbitrário. Sejam $\lambda_{min} < 0, \lambda_{max} > 0, \mu_{max} > 0$. Suponhamos que

$$\bar{\lambda}_i^1 \in [\lambda_{min}, \lambda_{max}] \; \forall \; i = 1, \dots, m$$

e

$$\bar{\mu}_i^1 \in [0, \mu_{max}] \; \forall \; i = 1, \dots, p.$$

Seja $\rho_1 > 0$. Definir $V^0 = g(x^0)_+$. Inicializar $k \leftarrow 1$.

Passo 1. Encontrar $x^k \in \Omega$, solução global do subproblema (3.1).

Passo 2. Definir

$$V_i^k = \max\left\{g_i(x^k), -\frac{\bar{\mu}_i^k}{\rho_k}\right\}, i = 1, \dots, p.$$

Se

$$\max\{\|h(x^k)\|_\infty, \|V^k\|_\infty\} \le \tau \; \max\{\|h(x^{k-1})\|_\infty, \|V^{k-1}\|_\infty\},$$

definir $\rho_{k+1} = \rho_k$. Em caso contrário, definir $\rho_{k+1} = \gamma \rho_k$.

Passo 3. Calcular

$$\bar{\lambda}_i^{k+1} = \min\{\lambda_{max}, \max\{\lambda_{min}, \bar{\lambda}_i^k + \rho_k h_i(x^k)\}\}$$

para $i = 1, \dots, m$, e

$$\bar{\mu}_i^{k+1} = \min\{\mu_{max}, \max\{0, \bar{\mu}_i^k + \rho_k g_i(x^k)\}\},$$

para $i = 1, \dots, p$.

Passo 4. Incrementar $k \leftarrow k+1$ e voltar ao Passo 1.

EXERCÍCIOS

1. Tente reproduzir os Argumentos do Teorema 2.3 para o Algoritmo 3.1. Onde a argumentação falha?

2. Suponha que x^k é gerado pelo Algoritmo 3.1 e que $h_i(x^k) = 0$, $g_j(x^k) = 0$ para todo $i = 1, \ldots, m, j = 1, \ldots, p$. (Ou seja, x^k é admissível e todas as restrições são ativas em x^k.) Prove que, neste caso, x^k é solução global de PNL.

3. A definição de V^k envolve a divisão de $\bar{\mu}^k$ pelo parâmetro de penalidade. Mas a motivação desta definição seria válida mesmo que $\bar{\mu}^k$ não fosse dividido por ρ_k. Encontre motivos para dividir por ρ_k.

4. Usando o programa `Algencan` do projeto TANGO, faça experimentos numéricos ilustrando as propriedades enunciadas neste capítulo.

5. Argumente sobre a definição de $\bar{\lambda}^{k+1}$ e $\bar{\mu}^{k+1}$ supondo que m e p podem ser maiores que 1. Encontre argumentos a favor e contra a posição de que "os multiplicadores de Lagrange somente devem ser atualizados nas iterações em que o parâmetro de penalidade não é aumentado".

6. Considere um PNL onde entre as restrições apareça o requisito de que certas variáveis devem ser inteiras. Expresse essas restrições de forma funcional adequada. Aplique `Algencan` e observe o comportamento. Analise problemas com solução conhecida e tire conclusões.

7. Há alguma relação entre $f(x^{k+1})$ e $f(x^k)$ no Método de Penalidade Deslocada? E entre os valores das funções objetivo de subproblemas consecutivos? E entre as medidas de não-admissibilidade? Prove ou dê contraexemplos.

8. Revise os resultados deste capítulo tentando eliminar a restrição de que $\bar{\lambda}^k$ e $\bar{\mu}^k$ descansem em conjuntos compactos. Mais precisamente, mostre que os mesmos resultados podem ser provados impondo apenas que $(\bar{\lambda}^k)^2/\rho_k$ e $(\bar{\mu}^k)^2/\rho_k$ tendam a zero sempre que ρ_k tende a infinito. Discuta se esta modificação vale a pena.

9. Considere o problema PNL e substitua cada restrição de desigualdade $g_i(x) \leq 0$ pela restrição de igualdade $g_i(x)+z_i^2 = 0$. Temos assim um problema apenas com restrições de igualdade. Defina o Algoritmo 3.3 para este novo problema. Mostre que ele coincide com o Algoritmo 3.3 para o problema original.

Capítulo 4

Condições de Otimalidade e Critérios de Parada

O problema geral de Otimização consiste em Minimizar $f(x)$ sujeita a $x \in D$. Neste livro D é um subconjunto de $\mathbb{R}^n$. Por ser esta definição tão simples, deveria ser redundante frisar que o que se deseja é encontrar $x^* \in D$ tal que $f(x^*) \leq f(x)$ para todo $x \in D$. Isto é, almejamos minimizadores *globais* do problema.

Entretanto, calcular soluções globais é, freqüentemente, muito difícil e, mesmo reconhecer se um ponto $x \in D$ é minimizador global resulta, em muitos casos, impossível. A maioria dos algoritmos populares de Otimização não garantem a descoberta de minimizadores globais. Paradoxalmente, é muito fácil definir algoritmos que, teoricamente, convergem a minimizadores globais. Basta usar qualquer procedimento que, em última análise, possa ser reduzido à procura exaustiva em um subconjunto denso de D. Tais métodos podem ser mais ou menos úteis mas sua eventual eficiência não é explicada por propriedades teóricas de convergência. Veja Exercício no final deste capítulo.

Aqui estamos interessados em algoritmos que –mais modestos– podem convergir ou não a minimizadores globais. Apesar desta fraqueza, o que pode ser provado sobre convergência destes algoritmos reflete bastante bem o que acontece na prática.

Formulemos, então, a pergunta: o que pode ser provado?

Como a propriedade de ser minimizador global é desejada, mas excessivamente ambiciosa, nos interessa analisar propriedades mais fracas que, no entanto, sejam satisfeitas pelos minimizadores globais do problema. Essas propriedades se denominam *Condições Necessárias de Otimalidade* (CNO) e os pontos que as satisfazem se dizem *estacionários*. Via de regra, usaremos a expressão *Condição de Otimalidade* como sinônimo de *Condição Necessária de Otimalidade*. Ou seja, uma condição de otimalidade se caracteriza pela implicação

$$\text{Minimizador global} \;\Rightarrow\; \text{Ponto estacionário.}$$

Entretanto, não é qualquer propriedade implicada pelo status de minimizador global que merece ser chamada Condição de Otimalidade. Com efeito, com caracterização tão formal do conceito de CNO, o fato de pertencer a D, ou mesmo de pertencer a $I\!R^n$ é uma condição de otimalidade. Porém, estas são condições excessivamente fracas e não merecem maior atenção. Assim, uma boa qualidade de condições necessárias de otimalidade é a de serem razoavelmente fortes.

Naturalmente, a CNO mais forte possível é a de ser minimizador global. Mas já mencionamos que conseguir pontos que a satisfaçam ou mesmo verificá-la num ponto dado é extremamente difícil. Um pouco mais manejável é a condição de minimizador local. Um ponto $x^* \in D$ é uma solução local (ou minimizador local) do problema de minimizar $f(x)$ sujeita a $x \in D$ se existe $\varepsilon > 0$ tal que $f(x^*) \leq f(x)$ para todo $x \in D$ cuja distância a x^* seja menor que ε. Certamente, a condição de minimizador local é estritamente mais fraca que a de minimizador global. Porém, ela não é totalmente satisfatória do ponto de vista algorítmico porque, em geral, tampouco é fácil de verificar.

Em última análise, gostariamos de uma CNO cuja verificação se limitasse a um cálculo mecânico. O exemplo mais familiar é o da minimização sem restrições ($D = I\!R^n$). Neste caso, se as derivadas parciais de f existem e x^* é um minimizador local, sabemos que $\nabla f(x^*) = 0$. (Ver Exercício neste capítulo.) Portanto, a verificação desta condição de otimalidade envolve apenas o cálculo das derivadas de f.

Podemos obter condições de otimalidade, razoavelmente fortes e fáceis de verificar, no problema de minimização *com* restrições?

Vamos nos concentrar no caso em que o problema de otimização é

$$\text{Minimizar } \ f(x) \text{ sujeita a } x \in D \tag{4.1}$$

com

$$D = \{x \in I\!R^n \mid h(x) = 0,\ g(x) \leq 0\}, \tag{4.2}$$

onde

$$h : I\!R^n \to I\!R^m, g : I\!R^n \to I\!R^p,$$

e as funções f, h e g são diferenciáveis em $I\!R^n$.

Se $p = 0$ temos o problema de minimização com restrições de igualdade, que aparece com freqüência nos cursos elementares de Cálculo em Várias Variáveis. Neste caso, se desenharmos (desenhe!) o conjunto D e um ponto x^* tal que $h(x^*) = 0$, podemos formular a pergunta: Como deveria ser $\nabla f(x^*)$ para que x^* fosse um minimizador local? É fácil intuír como $\nabla f(x^*)$ *não* deveria ser: Se $\nabla f(x^*)$ não fosse "perpendicular" ao conjunto D, o conjunto de nível de f atravessaria D e, portanto, deixaria de um lado pontos admissíveis arbitrariamente próximos de x^*, para os quais $f(x) < f(x^*)$. Isto parece indicar que $\nabla f(x^*)$ deveria ser perpendicular ao conjunto D no ponto x^*. Em outras palavras $\nabla f(x^*)$ deveria ser uma combinação linear dos gradientes das restrições $\nabla h_1(x^*), \ldots, \nabla h_m(x^*)$. Portanto, deveria existir $\lambda \in I\!R^m$ tal que

$$\nabla f(x^*) + \sum_{i=1}^{m} \lambda_i \nabla h_i(x^*) = 0.$$

Esta é a conhecida condição de Lagrange do cálculo elementar e os coeficientes λ_i se chamam *Multiplicadores de Lagrange.*

No caso em que $p > 0$, definindo

$$I_A(x) = \{i \in \{1, \ldots, p\} \mid g_i(x) = 0\},$$

podemos reproduzir os argumentos acima para chegar a que, além de $\lambda_1, \ldots, \lambda_m$, devem existir μ_i, $i \in I_A$, tais que

$$\nabla f(x^*) + \sum_{i=1}^{m} \lambda_i \nabla h_i(x^*) + \sum_{i \in I_A} \mu_i \nabla g_i(x^*) = 0.$$

Estendemos a denominação "Multiplicadores de Lagrange" para os coeficientes μ_i. Neste caso, podemos continuar o argumento gráfico-intuitivo para decidir se cada μ_i deve ser positivo ou negativo. Com efeito, se μ_i fosse menor que zero, ∇f e ∇g_i "apontariam para o mesmo lado". Portanto, quando g_i passasse de nulo a negativo, f diminuiria de valor. Ou seja, ao entrar na admissibilidade, f tomaria valores menores que em x^*. Isto não pode acontecer em um minimizador local. Logo, nos parece que os multiplicadores $\mu_i, i \in I_A$ devem ser não-negativos. Resumindo, temos bom motivos para acreditar que, se x^* é minimizador local de $f(x)$ sujeita a $h(x) = 0$, $g(x) \leq 0$, existem $\lambda \in I\!R^m$, $\mu \in I\!R^p_+$ tais que

$$\nabla f(x^*) + \nabla h(x^*)\lambda + \nabla g(x^*)\mu = 0,$$

onde

$$g_i(x^*) < 0 \Rightarrow \mu_i = 0.$$

Quando um ponto admissível satisfaz estas condições, dizemos que é um *Ponto KKT*. As condições KKT (Karush-Kuhn-Tucker) são a generalização para restrições de desigualdade das famosas condições de Lagrange. Observe que as condições KKT são bastante fáceis de verificar.

O argumento usado acima para motivar as condições KKT parece bastante plausível. Entretanto, não é verdade que as condições KKT sejam, rigorosamente falando, condições necessárias de otimalidade. De fato, nem as condições de Lagrange (caso particular) o são. Considere, por exemplo, o problema

$$\text{Minimizar } x_1 \text{ sujeita a } x_1^2 + x_2^2 = 0.$$

Claramente, $x^* = 0$ é o minimizador global do problema, mas as condições KKT não valem nesse ponto.

O que tem de estranho este probleminha? Antes de responder pediremos ao leitor que resolva o Exercício 3 da lista que segue ao presente capítulo.

Cumprida a tarefa, o leitor já deve ter percebido que, se m variáveis pudessem ser colocadas em evidência de maneira unívoca em uma vizinhança do minimizador x^*, as condições KKT seriam válidas. No exemplinho isso não pode ser feito. A condição suficiente mais popular para que m variáveis sejam isoláveis é que possa ser aplicado o Teorema da Função Implícita, o qual pode ser usado se os gradientes $\nabla h_1(x^*), \ldots, \nabla h_m(x^*)$ são linearmente independentes.

4.1 Condições de Qualificação

Terminamos a Seção anterior mostrando que, em um exemplo simples, as condições KKT falhavam em um minimizador local aparentemente porque os gradientes das restrições eram linearmente dependentes. Logo, parece sensato conjecturar que se os gradientes das restrições de igualdade junto com os gradientes das restrições onde $g_i(x^*) = 0$ são linearmente *independentes*, então as condições KKT valem no minimizador local. De agora em diante, as restrições de igualdade, junto com as restrições de desigualdade tais que $g_i(x) = 0$ no ponto admissível x serão chamadas *restrições ativas* em x. Quando os gradientes das restrições ativas em x são linearmente independentes, diremos que o ponto é *regular*.

Diremos que uma propriedade de pontos admissíveis da forma (4.2) é uma *Condição de Qualificação* (CQ) se o fato de ser satisfeita em um minimizador local x^* implica que x^* verifica as condições KKT.

Em outras palavras, se CQ é uma condição de qualificação, a proposição *KKT ou Não-CQ* é uma condição de otimalidade, isto é:

$$\text{Minimizador local} \;\Rightarrow\; (\text{ KKT ou Não-CQ})$$

qualquer que seja a função objetivo f. Portanto, assim como boas condições de otimalidade devem ser fortes, boas condições de qualificação devem ser fracas.

Voltemos à independência linear das restrições ativas (regularidade). Conjecturamos acima que esta propriedade tem boas chances de ser uma condição de qualificação. Mas, será que é a mais fraca possível? Antes de responder rigorosamente esta pergunta, pediremos ao leitor que examine as seguintes definições:

- Suponhamos que $h(x) = 0, g_i(x) \leq 0$, $I_A(x) = \{i \in \{1, \ldots, p\} \mid g_i(x) = 0\}$. Dizemos que x satisfaz a *Condição de Mangasarian-Fromovitz* (MF) se

  $$\sum_{i=1}^{m} \lambda_i \nabla h_i(x) + \sum_{i \in I_A} \mu_i \nabla g_i(x) = 0$$

  com $\mu_i \geq 0$ para todo $i \in I_A(x)$ implica que

  $$\lambda_1 = \ldots = \lambda_m = \mu_i = 0$$

para todo $i \in I_A$.

- Com a mesma definição de I_A, dizemos que x satisfaz a *Condição de Dependência Linear Positiva Constante* (CPLD) se a existência de $I_h \subset \{1, \ldots, m\}$, $I_g \subset I_A(x)$, $\lambda_i \in I\!R$ para todo $i \in I_h$, $\mu_i \geq 0$ para todo $i \in I_g$ tais que

$$\sum_{i \in I_h} \lambda_i \nabla h_i(x) + \sum_{i \in I_g} \mu_i \nabla g_i(x) = 0$$

com $\sum_{i \in I_h} |\lambda_i| + \sum_{i \in I_g} \mu_i > 0$, implica que os gradientes

$$\{\nabla h_i(z)\}_{i \in I_h}, \{\nabla g_i(z)\}_{i \in I_g}$$

são linearmente dependentes para todo z em uma vizinhança de x.

Observe que Mangasarian-Fromovitz implica CPLD e que a regularidade implica MF. Por outro lado, imagine um problema onde $m = 0, p = 2$, $g_1(x) = g_2(x)$ para todo x e $\nabla g_1(x^*) \neq 0$. Claramente, a condição MF se satisfaz em x^* mas a condição de independência linear dos gradientes das restrições ativas não se satisfaz. Por outra parte (desenhe!) é impossível que x^* seja minimizador local (independentemente de f!) sem que a condição KKT se satisfaça. Este exemplo parece sugerir que a regularidade não é a CQ mais fraca possível e que, talvez, MF também seja uma CQ.

Entretanto, considere um problema onde $m = 0, p = 2$, $g_1(x) = -g_2(x)$ para todo x e $\nabla g_1(x^*) \neq 0$. Neste caso, MF não se satisfaz (pois $\nabla g_1(x^*) + \nabla \underline{g}(x^*) = 0$) mas, obviamente, CPLD vale. De novo (desenhe!) é impossível que x^* seja minimizador local sem que se satisfaça a condição KKT.

Estas observações sugerem que CPLD é uma CQ. Ao mesmo tempo, como MF e a independência linear dos gradientes das restrições implicam CPLD, as mesmas também seriam condições (mais fortes) de qualificação. Uma propriedade importante da CPLD é que esta condição se cumpre sempre que as restrições são lineares.

4.2 CPLD é uma Condição de Qualificação

O título desta seção expressa totalmente seu objetivo. Forneceremos uma prova rigorosa de que CPLD é uma condição de qualificação. A ferramenta fundamental para isto será o Método de Penalidade Externa introduzido no Capítulo 2. Um ingrediente auxiliar importante será o seguinte Lema de Carathéodory.

Lema 4.1. *Suponhamos que*

$$u = \sum_{i=1}^{m} \lambda_i v^i + \sum_{j=1}^{p} \mu_j w^j,$$

onde $v^i \in I\!R^n, w^j \in I\!R^n$ *para todo* $i = 1, \ldots, m$, $j = 1, \ldots, p$, *e* $\mu_j \geq 0$ *para todo* $j = 1, \ldots, p$. *Então, existem* $I \subset \{1, \ldots, m\}, J \subset \{1, \ldots, p\}$, $\{\lambda'_i\}_{i \in I} \subset I\!R$, $\{\mu'_j\}_{j \in J} \subset I\!R_+$, *tais que*

$$u = \sum_{i \in I} \lambda'_i v^i + \sum_{j \in J} \mu'_j w^j$$

e os vetores $\{v^i\}_{i \in I} \cup \{w^j\}_{j \in J}$ *são linearmente independentes.*

Prova. Sem perda de generalidade, suponhamos que $\lambda_i \neq 0, \mu_j > 0$ para todo $i = 1, \ldots, m, j = 1, \ldots, p$.

Suponhamos que os vetores $\{v^1, \ldots, v^m, w^1, \ldots, w^p\}$ são linearmente dependentes. Portanto, existem escalares α_i e β_j, não todos nulos, tais que:

$$0 = \sum_{i=1}^{m} \alpha_i v^i + \sum_{j=1}^{p} \beta_j w^j.$$

Logo, pela hipótese, temos que

$$u = \sum_{i=1}^{m} (\lambda_i - t\alpha_i) v^i + \sum_{j=1}^{p} (\mu_j - t\beta_j) w^j$$

para todo $t \in I\!R$. Para $t = 0$, nenhum dos coeficientes na igualdade acima se anula. Seja t_{min} o t de menor módulo que anula pelo menos um dos coeficientes $\lambda_i - t\alpha_i$ ou $\mu_j - t\beta_j$. Então:

$$u = \sum_{i=1}^{m} (\lambda_i - t_{min}\alpha_i) v^i + \sum_{j=1}^{p} (\mu_j - t_{min}\beta_j) w^j.$$

Claramente, $\mu_j - t_{min}\beta_j \geq 0$ para todo j, mas conseguimos escrever u como combinação linear de, no máximo, $m + p - 1$ vetores. Este processo pode ser repetido até que os vetores que contribuem à combinação linear sejam todos linearmente independentes. **QED**

Os dois resultados principais seguem em continuação. O segundo afirma que, com efeito, a CPLD é uma CQ, o que implica que regularidade e MF também o são. Daremos por conhecido o fato de que, se uma função tem um minimizador local em um ponto interior admissível onde as derivadas parciais existem, então estas derivadas devem ser zero nesse ponto. Veja Exercício 1 deste capítulo.

Teorema 4.1. *Seja x^* minimizador local do problema (4.1)-(4.2) onde as funções f, h, g são deriváveis e contínuas. Então, existem seqüências $\{x^k\} \subset I\!R^n$, $\{\lambda^k\} \subset I\!R^m$, $\{\mu^k\} \subset I\!R^p_+$, tais que:*

- $\lim_{k\to\infty} x^k = x^*$,
- $\lim_{k\to\infty} \nabla f(x^k) + \nabla h(x^k)\lambda^k + \nabla g(x^k)\mu^k = 0$,
- $\lim_{k\to\infty} \|h(x^k)\| = \lim_{k\to\infty} \|g(x^k)_+\| = 0$,
- *Para todo $i = 1, \ldots, p$ e para todo $k \in I\!N$, $g_i(x^k) < 0 \Rightarrow \mu_i^k = 0$.*

Prova. Por ser minimizador local, existe $\delta > 0$ tal que x^* é solução global de

$$\text{Minimizar } f(x)$$

$$\text{sujeita a } h(x) = 0,\ g(x) \leq 0,\ \|x - x^*\| \leq \delta.$$

Portanto, x^* é a única solução global do problema:

$$\text{Minimizar } f(x) + \frac{1}{2}\|x - x^*\|^2 \quad \text{s.a. } h(x) = 0,\ g(x) \leq 0,\ \|x - x^*\| \leq \delta. \tag{4.3}$$

Consideremos a aplicação do Método de Penalidade Externa a (4.3). Cada subproblema consistirá em:

$$\text{Minimizar } f(x) + \frac{1}{2}\|x - x^*\|^2 + \frac{\rho_k}{2}[\|h(x)\|^2 + \|g(x)_+\|^2]$$

$$\text{sujeita a } \|x - x^*\| \leq \delta.$$

Cada um destes subproblemas consiste em minimizar uma função contínua em um compacto, portanto sua solução global x^k existe para todo $k \in I\!N$. Como x^* é a única solução global de (4.3), o Teorema de Convergência do Método de Penalidade Externa garante que:

$$\lim_{k\to\infty} x^k = x^*.$$

Logo, $\|x^k - x^*\| < \delta$ para k suficientemente grande, ou seja, x^k é um ponto interior do subproblema. Portanto, o gradiente da função objetivo deste subproblema deve ser zero. Isto é, para todo $k \in I\!N$ suficientemente grande,

$$\nabla f(x^k) + \nabla h(x^k)[\rho_k h(x^k)] + \nabla g(x^k)[\rho_k g(x^k)_+] + (x^k - x^*) = 0.$$

Definamos $\lambda^k = \rho_k h(x^k)$, $\mu^k = \rho_k g(x^k)_+$. Como $\lim_{k\to\infty} \|x^k - x^*\| = 0$, obtemos:

$$\lim_{k\to\infty} \nabla f(x^k) + \nabla h(x^k)\lambda^k + \nabla g(x^k)\mu^k = 0.$$

Agora, como $\mu^k = \rho_k g(x^k)_+$, temos que $\mu_i^k = 0$ se $g_i(x^k) < 0$. **QED**

Corolário 4.1 *Suponhamos as hipóteses do Teorema 4.1. Seja $\varepsilon > 0$. Então, existem $x \in I\!R^n$, $\lambda \in I\!R^m$, $\mu \in I\!R^p_+$ tais que:*

- $\|\nabla f(x) + \nabla h(x)\lambda + \nabla g(x)\mu\| \leq \varepsilon$,
- $\|h(x)\| \leq \varepsilon$, $\|g(x)_+\| \leq \varepsilon$,
- *Para todo* $i = 1, \ldots, p$, $g_i(x) < 0 \Rightarrow \mu_i = 0$.

Prova. Segue imediatamente do Teorema 4.1. **QED**

Em continuação, provamos que, se um minimizador local x^* satisfaz a condição CPLD, então deve satisfazer as condições KKT. Em outras palavras, a CPLD é uma condição de qualificação.

Teorema 4.2. *Consideremos o problema (4.1)-(4.2), onde as funções f, h, g são deriváveis e as derivadas são contínuas no minimizador local x^*. Suponhamos que x^* satisfaz a condição CPLD. Então, existem $\lambda \in I\!R^m$, $\mu \in I\!R^p_+$, tais que:*

- $\nabla f(x^*) + \nabla h(x^*)\lambda + \nabla g(x^*)\mu = 0$,
- *Para todo* $i = 1, \ldots, p$, $g_i(x^*) < 0 \Rightarrow \mu_i = 0$.

Prova. Pelo Teorema 4.1, existem seqüências $\{x^k\} \subset I\!R^n$, $\{\lambda^k\} \subset I\!R^m$, $\{\mu^k\} \subset I\!R^p_+$, tais que:

- $\lim_{k\to\infty} x^k = x^*$,
- $\lim_{k\to\infty} \nabla f(x^k) + \nabla h(x^k)\lambda^k + \nabla g(x^k)\mu^k = 0$,
- $\lim_{k\to\infty} h(x^k) = \lim_{k\to\infty} g(x^k)_+ = 0$,
- Para todo $i = 1, \ldots, p$, $g_i(x^k) < 0 \Rightarrow \mu_i^k = 0$.

Portanto, para todo $k \in I\!N$, existe $\epsilon^k \in I\!R^n$ tal que

$$-\nabla f(x^k) + \epsilon^k = \nabla h(x^k)\lambda^k + \nabla g(x^k)\mu^k$$

e $\epsilon^k \to 0$.

Pelo Lema 4.1, temos que:

$$-\nabla f(x^k) + \epsilon^k = \sum_{i \in I_k} \widehat{\lambda}_i^k \nabla h_i(x^k) + \sum_{j \in J_k} \widehat{\mu}_j^k \nabla g_j(x^k),$$

onde $\epsilon^k \to 0$, os vetores $\{\nabla h_i(x^k)\}_{i \in I_k} \cup \{\nabla g_j(x^k)\}_{j \in J_k}$ são linearmente independentes, e $g_j(x^k) \geq 0$ sempre que $j \in J_k$.

Como o número de possíveis subconjuntos I_k, J_k é finito, existe $K \underset{\infty}{\subset} I\!N$ tal que para todo $k \in K$, temos que $I_k = I \subset \{1, \ldots, m\}$ e $J_k = J \subset \{1, \ldots, p\}$. Logo, para todo $k \in K$,

$$-\nabla f(x^k) + \epsilon^k = \sum_{i \in I} \widehat{\lambda}_i^k \nabla h_i(x^k) + \sum_{j \in J} \widehat{\mu}_j^k \nabla g_j(x^k),$$

onde $\epsilon^k \to 0$ e os vetores $\{\nabla h_i(x^k)\}_{i \in I} \cup \{\nabla g_j(x^k)\}_{j \in J}$ são linearmente independentes.

Portanto,

$$\lim_{k \in K} \nabla f(x^k) + \sum_{i \in I} \widehat{\lambda}_i^k \nabla h_i(x^k) + \sum_{j \in J} \widehat{\mu}_j^k \nabla g_j(x^k) = 0, \tag{4.4}$$

onde $g_j(x^k) \geq 0$ sempre que $j \in J$.

Agora consideremos dois casos. Se as seqüências $\{\widehat{\lambda}_i^k\}$ e $\{\widehat{\mu}_j^k\}$ estão contidas em um compacto para todo $i \in I$, $j \in J$, a tese segue passando ao limite em uma subseqüência convergente. Em caso contrário, definimos, para todo k,

$$M_k = \max\{|\lambda_i^k|, \mu_j^k, i \in I, j \in J\}.$$

Neste caso, existe $K_1 \underset{\infty}{\subset} K$ tal que $\lim_{k \in K_1} M_k = \infty$. Logo, por (4.4),

$$\lim_{k \in K_1} \frac{\nabla f(x^k)}{M_k} + \sum_{i \in I} \nabla h_i(x^k) \frac{\widehat{\lambda}_i^k}{M_k} + \sum_{j \in J} \nabla g_j(x^k) \frac{\widehat{\mu}_j^k}{M_k} = 0.$$

Portanto,

$$\lim_{k \in K_1} \sum_{i \in I} \nabla h_i(x^k) \frac{\widehat{\lambda}_i^k}{M_k} + \sum_{j \in J} \nabla g_j(x^k) \frac{\widehat{\mu}_j^k}{M_k} = 0.$$

Como o número de restrições $(m + p)$ é finito, existem um índice que realiza o máximo M_k infinitas vezes, digamos, para todo $k \in K_2 \underset{\infty}{\subset} K_1$. Mais ainda, existe

$K_3 \underset{\infty}{\subset} K_2$ tal que as seqüências $\frac{\widehat{\lambda}^k}{M_k}$ e $\frac{\widehat{\mu}^k}{M_k}$ convergem a um conjunto de coeficientes não todos nulos, já que pelo menos um dos coeficientes tem módulo sempre igual a 1. Logo, tomando limites para $k \in K_3$, obtemos que os gradientes $\{\nabla h_i(x^*)\}_{i \in I} \cup \{\nabla g_j(x^*)\}_{j \in J}$ são linearmente dependentes com coeficientes não-negativos correspondendo a cada $\nabla g_j(x^*)$. Mas os vetores $\{\nabla h_i(x^k)\}_{i \in I} \cup \{\nabla g_j(x^k)\}_{j \in J}$ são linearmente independentes e a seqüência $\{x^k\}$ converge a x^*, portanto x^* não satisfaz a condição CPLD, contrariamente ao suposto na hipótese do teorema. **QED**

4.3 Critérios de Parada

Todos os algoritmos para resolver problemas não triviais de Otimização são iterativos. Reconhecer se um iterando é, ou está perto, de um minimizador global é, em geral, muito difícil, exceto em situações especiais, por exemplo, quando o valor da função objetivo no minimizador global é conhecido. Ao mesmo tempo, a maioria dos algoritmos eficientes não possuem a propriedade de convergir a minimizadores globais. Por isso, para *parar* a execução de um algoritmo se usam as condições necessárias de otimalidade, ou versões aproximadas das mesmas.

Como vimos na seção anterior, as condições necessárias de otimalidade tem a forma

$$KKT \text{ ou } \text{ Não-CQ } .$$

Isto parece indicar que um algoritmo deveria parar no iterando (aproximadamente admissível) x^k se as condições KKT fossem aproximadamente satisfeitas ou uma Condição de Qualificação fosse "aproximadamente não-satisfeita". Para fixar idéias, pensemos na regularidade, que é a condição de qualificação mais simples. De acordo com o indicado acima, se x^k é aproximadamente admissível, deveriamos parar o algoritmo nesse ponto se uma das seguintes propriedades fosse satisfeita:

- O ponto x^k satisfaz, aproximadamente, as condições KKT.
- Os gradientes das restrições ativas em x^k são, aproximadamente, linearmente dependentes.

Entretanto, embora todos os algoritmos conhecidos testem a admissibilidade e as condições KKT nos iterandos x^k, dificilmente algoritmos eficientes param por encontrar gradientes de restrições quase dependentes. Muito menos os algoritmos se preocupam por testar outras condições de qualificação como Mangasarian-Fromovitz ou CPLD.

O fato é que eles "têm razão", e a chave da racionalidade de não se preocupar pelas Condições de Qualificação está expressa no Corolário 4.1. Este corolário diz que em qualquer vizinhança de um minimizador local podem ser encontrados pontos que, aproximadamente, são admissíveis e KKT! Ou seja, a presença de pontos assim em toda vizinhança de x^* é uma condição necessária de otimalidade.

Temos, assim, a tentação de postular que o critério de convergência prático para qualquer algoritmo destinado a resolver (4.1)–(4.2) deve ser a existência de multipicadores $\lambda \in I\!R^m$ e $\mu \in I\!R^p_+$, associados ao iterando x, tais que

- $\|\nabla f(x) + \nabla h(x)\lambda + \nabla g(x)\mu\| \leq \varepsilon$,
- $\|h(x)\| \leq \varepsilon$, $\|g(x)_+\| \leq \varepsilon$,
- $g_i(x) < 0 \Rightarrow \mu_i = 0$ para todo $i = 1, \ldots, p$

Entretanto, vejamos em um exemplo simples que a terceira condição acima, chamada de *complementaridade*, é demasiado exigente e pode não ser satisfeita por pontos arbitrariamente próximos a um minimizador. Consideremos o problema

$$\text{Minimizar } x \text{ sujeita a } -x \leq 0,$$

cuja solução global, óbvia, é $x^* = 0$. Suponhamos que um bom algoritmo produz a seqüência $\{x^k\}$, que converge a x^*, com todos seus elementos estritamente maiores que zero. Portanto, para todo x^k, se a complementaridade fosse satisfeita, o multiplicador μ associado deveria ser zero. Mas, neste caso, $\nabla f(x) = 1$ para todo x, portanto a desigualdade $\|\nabla f(x) + \nabla h(x)\lambda + \nabla g(x)\mu\| \leq \varepsilon$ não se cumpriria para nenhum $\varepsilon < 1$.

A necessidade de relaxar a condição de complementaridade nos leva à seguinte definição:

Diremos que um ponto $x \in I\!R^n$ é ε-*KKT* do problema (4.1)-(4.2) se existem $\lambda \in I\!R^m$, $\mu \in I\!R^p_+$ tais que:

- $\|\nabla f(x) + \nabla h(x)\lambda + \nabla g(x)\mu\| \leq \varepsilon$,
- $\|h(x)\| \leq \varepsilon$, $\|g(x)_+\| \leq \varepsilon$,
- $g_i(x) < -\varepsilon \Rightarrow \mu_i = 0$ para todo $i = 1, \ldots, p$.

Diremos, ao mesmo tempo, que x^* é um ponto AKKT (aproximadamente KKT) se, para todo $\varepsilon > 0$, existe um ponto ε-KKT cuja distância a x^* é menor que ε.

Pelo Corolário 4.1, todo minimizador local é AKKT. Ou seja, todo minimizador local é aproximável por pontos ε-KKT. Isto significa que a existência de pontos ε-KKT arbitrariamente próximos a x^* é uma condição necessária de otimalidade. Por outro lado, ela é mais razoável que a sugerida pela tese do Corolário 4.1 no sentido de que a complementaridade não é exigida para pontos admissíveis muito próximos de x^*. É fácil ver que esta condição não falha em pontos muito próximos de $x^* = 0$ no exemplo simples mencionado acima.

Resta se perguntar se AKKT é uma condição de otimalidade razoavelmente forte. Uma resposta vem dada pelo seguinte teorema, onde mostramos que a presença de uma seqüência de pontos ε_k-KKT, com $\varepsilon_k \to 0$ também implica a condição *KKT ou Não-CPLD.*

Teorema 4.3. *Consideremos o problema (4.1)-(4.2), onde as funções f, h, g são deriváveis e as derivadas são contínuas em x^*. Suponhamos que existem seqüências $\{x^k\} \subset I\!R^n$, $\{\lambda^k\} \subset I\!R^m$, $\{\mu^k\} \subset I\!R^p_+$, $\{\varepsilon_k\} \subset I\!R_{++}$ tais que*

- $\|\nabla f(x^k) + \nabla h(x^k)\lambda^k + \nabla g(x^k)\mu^k\| \leq \varepsilon_k$,
- $\|h(x^k)\| \leq \varepsilon_k$, $\|g(x^k)_+\| \leq \varepsilon_k$,
- $g_i(x^k) < -\varepsilon_k \Rightarrow \mu_i^k = 0$ *para todo* $k \in I\!N$, $i = 1, \ldots, p$
- $\lim_{k\to\infty} x^k = x^*$.
- $\lim_{k\to\infty} \varepsilon_k = 0$.

Então:

- x^* *é admissível;*
- *Se x^* satisfaz a CPLD, existem $\lambda \in I\!R^m$, $\mu \in I\!R^p_+$ tais que*

$$\nabla f(x^*) + \nabla h(x^*)\lambda + \nabla g(x^*)\mu = 0$$

  e

$$\mu_i = 0 \text{ para todo } i \text{ tal que } g_i(x^*) < 0.$$

Prova. Percorra a prova do Teorema 4.2. Observe que todos os argumentos desta prova valem neste caso, com os seguintes destaques:

$$j \in J_k \Rightarrow g_j(x^k) \geq -\varepsilon_k,$$

$$j \in J \Rightarrow g_j(x^k) \geq -\varepsilon_k.$$

Como conseqüência, se $j \in J$ e $x^k \to x^*$, temos que $g_j(x^*) \geq 0$. Isto implica que a condição de quase-complementaridade $g_i(x^k) < -\varepsilon_k \Rightarrow \mu_i^k = 0$ é suficiente para garantir a complementaridade no limite. **QED**

EXERCÍCIOS

1. Provar que, se $f : D \to I\!R$, x^* é um ponto interior de D, x^* é minimizador local e $\nabla f(x^*)$ existe, então $\nabla f(x^*) = 0$.

2. A razão principal pela qual as condições *suficientes* de otimalidade usuais são menos importantes que as condições necessárias é que tais condições costumam ser suficientes para minimizador local, mas não para minimizador global. Portanto, procurando condições suficientes, os minimizadores globais podem escapar. Encontre exemplos de minimização sem restrições (mesmo em uma variavel) onde o minimizador global não satisfaça a condição suficiente usual para minimizador local (Hessiana definida positiva).

3. Considere o problema de minimizar $f(x)$ sujeita a $h(x) = 0$ e suponha que, no sistema de equações $h(x) = 0$, m variáveis podem ser postas em evidência em função das restantes $n - m$ variáveis, ao menos em uma vizinhança do minimizador local x^*. Prove que, nesse caso, as condições KKT (ou Lagrange) se satisfazem. Faça este exercício sem usar os teoremas provados neste capítulo.

4. Usando o Lema de Carathéodory, prove que o fecho convexo de um conjunto compacto deve ser compacto.

5. Prove que, se a propriedade B é uma CQ e A implica B, então a propriedade A é uma CQ.

6. Prove que Regularidade $\Rightarrow$ MF $\Rightarrow$ CPLD. Analise cada uma destas condições desde o ponto de vista de facilidade de verificação.

7. Neste capítulo consideramos que a região admissível do problema central vem dada por $h(x) = 0, g(x) \leq 0$. Suponha que, além destas restrições temos $x \in \Omega$ e prove os resultados do capítulo com a hipótese de que o minimizador local x^* é um ponto interior de Ω.

8. Prove que se x^* é minimizador local e satisfaz MF, então o conjunto de multiplicadores associados a x^* é compacto e convexo.

9. No começo deste capítulo declaramos ser muito fácil definir algoritmos (ineficientes) tais que todo ponto de acumulação é um minimizador global. Defina um algoritmo desse tipo rigorosamente e medite sobre o assunto.

10. Considere o problema de minimizar $f(x, y)$ sujeita a $xy = 0$, $x \geq 0, y \geq 0$. Verifique que CPLD não é satisfeita em nenhum ponto viável (portanto LIAC e Mangasarian-Fromovitz tampouco). Que acontece com a condição de otimalidade que chamamos AKKT?

11. Analise as diferentes condições necessárias de otimalidade desde o ponto de vista da maior ou menor facilidade de verificação.

12. Invente problemas simples (digamos, com $n = 2$) com diferentes condições de qualificação. Execute `Algencan` para resolver esses problemas, observe e interprete o comportamento do algoritmo. Escreva uma pequena monografia a respeito.

13. Diremos que um ponto $x \in I\!R^n$ é ε-*KKT* forte do problema (4.1)-(4.2) se existem $\lambda \in I\!R^m$, $\mu \in I\!R^p_+$ tais que:

    - $\|\nabla f(x) + \nabla h(x)\lambda + \nabla g(x)\mu\| \leq \varepsilon$,
    - $\|h(x)\| \leq \varepsilon$, $\|g(x)_+\| \leq \varepsilon$,
    - $g_i(x) < 0 \Rightarrow \mu_i = 0$. para todo $i = 1, \ldots, p$

    Diremos, ao mesmo tempo, que x^* é um ponto AKKT-forte se, para todo $\varepsilon > 0$, existe um ponto ε-KKT forte cuja distância a x^* é menor que ε. Obviamente,
    $$\varepsilon\text{-KKT forte} \Rightarrow \varepsilon\text{-KKT}$$
    e
    $$\text{AKKT-forte} \ \Rightarrow \ \text{AKKT}.$$

    Investigue a seguinte questão: A condição AKKT-forte é *estritamente* mais forte que AKKT?

14. Se diz que o ponto admissível x^* satisfaz a condição AGP(γ) (onde $\gamma \in (0, \infty]$) (introduzida por Martínez e Svaiter, 2003) se existe uma seqüência $\{x^k\}$ que converge a x^* e tal que $\|P_{A_k}(x^k - \nabla f(x^k)) - x^k\| \to 0$, onde P_{A_k} é a projeção em A_k e A_k é o poliedro definido por:
    $$\nabla h(x^k)^T(x - x^k) = 0,$$
    $$\nabla g_i(x^k)^T(x - x^k) \leq 0 \text{ se } g_i(x^k) \geq 0,$$
    $$g_i(x^k) + \nabla g_i(x^k)^T(x - x^k) \leq 0 \text{ se } 0 > g_i(x^k) \geq -\gamma.$$

    Interprete geometricamente esta condição. Prove que todo minimizador local é AGP. Estabeleça e prove todas as relações possíveis entre AKKT e AGP.

15. Prove que se um ponto KKT satisfaz Mangasarian-Fromovitz, o conjunto dos multiplicadores de Lagrange associado é limitado e convexo.

Capítulo 5

Subproblemas Irrestritos

Em cada iteração do Método de Lagrangiano Aumentado precisamos minimizar a função $L_\rho(x, \lambda, \mu)$ com respeito a x, em um conjunto geralmente simples que chamamos Ω. Neste capítulo consideramos o caso menos complicado: $\Omega = I\!R^n$. Portanto, em cada iteração precisamos resolver um subproblema de minimização sem restrições. Por tradição, e para simplificar a notação, chamaremos f a função objetivo deste subproblema.

Em princípio, vamos supor que f tem derivadas primeiras contínuas para todo $x \in I\!R^n$. Nada diremos, no começo, sobre derivadas segundas. Esta omissão é conveniente já que a função Lagrangiano Aumentado não tem derivadas segundas contínuas quando o PNL tem desigualdades, independentemente do grau de suavidade das funções que as definem.

5.1 Algoritmo Geral

Definiremos um algoritmo bastante geral para minimização sem restrições, dentro do qual se encaixa a maioria dos algoritmos eficientes conhecidos. Como todos os métodos para resolver problemas não-lineares, trata-se de um algoritmo iterativo, que gera uma seqüência $\{x^k\} \subset I\!R^n$ partindo de um ponto inicial x^0.

Algoritmo 5.1. Sejam $\theta \in (0, 1), \alpha \in (0, 1/2), M > 1, \beta > 0$ os parâmetros algorítmicos. Seja $x^0 \in I\!R^n$ o ponto inicial. Dado $x^k \in I\!R^n$, os passos para calcular x^{k+1} são:

Passo 1. Se $\|\nabla f(x^k)\| = 0$, terminar a execução do algoritmo.

Passo 2. Calcular $d^k \in I\!R^n$ tal que

$$\nabla f(x^k)^T d^k \leq -\theta \|d^k\| \|\nabla f(x^k)\| \quad \text{e} \quad \|d^k\| \geq \beta \|\nabla f(x^k)\|.$$

Passo 3. Calcular $t_k > 0$, $x^{k+1} \in I\!R^n$, tal que

$$f(x^{k+1}) \leq f(x^k + t_k d^k) \leq f(x^k) + \alpha t_k \nabla f(x^k)^T d^k$$

e

$$\Big[t_k \geq 1\Big] \quad \text{ou} \quad \Big[f(x^k + \bar{t}_k d^k) > f(x^k) + \alpha \bar{t}_k \nabla f(x^k)^T d^k \;\; \text{para algum } \bar{t}_k \leq M t_k\Big].$$

Vejamos as razões que fundamentam cada passo deste método.

1. No Passo 1 decretamos que, se o gradiente no ponto atual x^k é nulo, não vale a pena continuar a execução do algoritmo. Ou seja, nos conformamos com encontrar pontos onde o gradiente é zero e não pretendemos ir mais longe que isso.

2. Se o gradiente não se anula em x^k, procuramos uma *direção de busca* d^k no Passo 2. Duas condições são exigidas a uma direção desse tipo. A primeira é que deve ser uma *direção de descida*, isto é, a derivada direcional $\nabla f(x^k)^T d^k$ deve ser menor que zero. Mais precisamente, o ângulo entre d^k e $-\nabla f(x^k)$ deve ser menor ou igual que um ângulo fixo menor que $\pi/2$, definido pelo parâmetro algorítmico θ. Se θ fosse igual a 1, a direção d^k deveria ser um múltiplo do gradiente. Em geral somos muito menos exigentes, e $\theta = 10^{-6}$ é a tolerância recomendada pela tradição. A segunda condição é que o tamanho de d^k deve ser, pelo menos, proporcional ao tamanho de $\nabla f(x^k)$. A constante de proporcionalidade é β. A razão para isto é que, se $\nabla f(x^k)$ é grande, provavelmente estamos longe da solução e, em conseqüência, queremos uma direção de busca grande.

3. No Passo 3 estabelecemos que o ponto final de nossa busca linear, $x^k + t_k d^k$, deve satisfazer a *Condição de Armijo*:

   $$f(x^k + t_k d^k) \leq f(x^k) + \alpha t_k \nabla f(x^k)^T d^k.$$

   Se definimos $\varphi(t) = f(x^k + t d^k)$, a condição de Armijo equivale a

   $$\varphi(t_k) \leq \varphi(0) + \alpha t_k \varphi'(0).$$

   Em outras palavras, com esta condição se exige que $\varphi(t_k)$ fique por baixo da reta que passa por $(0, \varphi(0))$ e tem inclinação $\alpha\varphi'(0)$. Pretendemos assim, não apenas que $f(x^k + t_k d^k)$ seja menor que $f(x^k)$, mas que seja "suficientemente menor". Por isso esta condição se denomina, muitas vezes, de *decréscimo suficiente*.

Entretanto, não nos satisfazemos com que a Condição de Armijo se verifique. Temos de garantir que o passo t_k só não é maior porque tal coisa é impossível, ou seja, que não tomamos passos artificialmente pequenos. Por isso, exigimos que, ou $t_k \geq 1$, ou há um passo fracassado $\bar{t}_k$, não muito maior que t_k, onde a Condição de Armijo não se verificou.

Por último, desejamos deixar a porta aberta para escolher um ponto x^{k+1} que seja, inclusive, melhor que o próprio ponto $x^k + t_k d^k$. Por isso impomos, para x^{k+1}, a mera exigência de que seu valor funcional seja menor ou igual a $f(x^k + t_k d^k)$. Obviamente, é admissível definir $x^{k+1} = x^k + t_k d^k$.

O Passo 3 do Algoritmo 5.1 admite muitas implementações diferentes, algumas boas, outras não. A estratégia mais elementar consiste em escolher t_k como o primeiro número da seqüência $\{2^{-\ell}\}$ que satisfaz a Condição de Armijo. É fácil ver que, dessa maneira, se obtém um passo de acordo com o estabelecido no algoritmo, com $M = 2$. A direção d^k também pode ser escolhida de muitas maneiras e, neste caso, a mais óbvia é $d^k = -\nabla f(x^k)$. Com essas duas escolhas, obtemos uma versão do *Método de Máxima Descida*, um dos procedimentos mais conhecidos para minimizar sem restrições.

O Algoritmo 5.1 tem quatro parâmetros algorítmicos: θ, α, M e β. Os três primeiros são adimensionais, isto é, servem para calcular magnitudes do mesmo tipo e, portanto, seu valor recomendado pode ser discutido independentemente dos dados do problema. Por exemplo, θ é o mínimo valor permitido para o cosseno de um ângulo, e α é a fração da derivada direcional que define o decréscimo suficiente. Neste caso, a tradição é usar $\alpha = 10^{-4}$.

O valor de M se relaciona com a estratégia usada na busca linear para retroceder quando o decréscimo suficiente não se verifica para um passo $\bar{t}$. O usual, neste caso, é escolher o novo passo, menor que $\bar{t}$, como o minimizador de uma parábola (ou cúbica) que interpola φ entre $t = 0$ e $t = \bar{t}$. Mas tal minimizador (t_{min}) poderia ficar muito próximo de 0 ou muito próximo de $\bar{t}$, com o qual gastar uma avaliação em $x^k + t_{min} d^k$ seria pouco econômico. Por isso se usam *salvaguardas*. Se t_{min} é menor que $\bar{t}/M$ ou maior que $\bar{t}(1 - 1/M)$, este passo é descartado e substituído por $t_{min} = \bar{t}/2$. Com isto, o segundo preceito do Passo 3 do algoritmo é obedecido.

O parâmetro algorítmico mais delicado é β porque, ao contrário dos outros, é *dimensional*, isto é, compara magnitudes de diferente tipo. Com efeito, suponhamos que, em vez de minimizar a função f, temos de minimizar a função $\bar{f} = 10f$. Os dois problemas são totalmente equivalentes e, portanto, desejaríamos que o algoritmo de minimização tivesse idêntico comportamento em ambos casos. Entretanto, $\nabla \bar{f} = 10 \nabla f$ e, portanto, as condições $\|d^k\| \geq \beta \|\nabla f(x^k)\|$ e $\|d^k\| \geq \beta \|\nabla \bar{f}(x^k)\|$ não são equivalentes. Em outras palavras, deveriam ser usados diferentes valores de β nos dois problemas. Uma análise mais cuidadosa revelaria que o valor de β deve ser proporcional à norma da inversa da Hessiana em x^k. Felizmente, os procedimentos usados para calcular d^k frequentemente satisfazem a condição $\|d^k\| \geq \beta\|\nabla f(x^k)\|$ automaticamente, para um valor de β que não conheceremos, mas pelo qual não precisaremos nos preocupar.

Em `Gencan`, que é o método para resolver subproblemas simples usado em `Algencan`, a direção d^k é calculada por uma estratégia chamada *Newton-Truncado* ou, alternativamente, por uma estratégia *quase-Newton*. O Passo 3 também é implementado de maneira sofisticada, com a inclusão de interpolações quando é necessário diminuír t e de extrapolações, quando parece que é conveniente avançar.

5.2 Boa Definição e Convergência Global

Veremos primeiro que o Algoritmo 5.1 está bem definido. Isto significa que, se o algoritmo não pára em x^k, sempre é possível obter x^{k+1}.

Teorema 5.1. *O Algoritmo 5.1 está bem definido e termina em x^k se, e somente se, $\nabla f(x^k) = 0$.*

Prova. Suponhamos que $\nabla f(x^k) \neq 0$. Pelas condições do Passo 2 e a diferenciabilidade de f,

$$\lim_{t\to 0} \frac{f(x^k + td^k) - f(x^k)}{t} = \nabla f(x^k)^T d^k < 0.$$

Logo,

$$\lim_{t\to 0} \frac{f(x^k + td^k) - f(x^k)}{t\nabla f(x^k)^T d^k} = 1.$$

Como $\alpha < 1$, para t suficientemente pequeno, temos:

$$\frac{f(x^k + td^k) - f(x^k)}{t\nabla f(x^k)^T d^k} \geq \alpha.$$

Como $\nabla f(x^k)^T d^k < 0$, deduzimos:

$$f(x^k + td^k) \leq f(x^k) + \alpha t \nabla f(x^k)^T d^k.$$

para t suficientemente pequeno. Logo, se escolhemos t_k como o primeiro número da seqüência $\{M^{-\ell}\}$ que satisfaz a condição acima, as condições do Passo 3 do Algoritmo 5.1 ficam satisfeitas. **QED**

O seguinte teorema se diz *de Convergência Global*. Ele estabelece que, independentemente do ponto inicial, se houver pontos limite, o gradiente deve se anular nos mesmos.

Teorema 5.2. *Se x^* é um ponto limite de uma seqüência gerada pelo Algoritmo 5.1, então $\nabla f(x^*) = 0$.*

Prova. Seja $K = \{k_0, k_1, k_2, k_3, \ldots\} \underset{\infty}{\subset} I\!N$ uma seqüência de inteiros tais que

$$\lim_{k \in K} x^k = x^*.$$

Pela continuidade de f,

$$\lim_{k \in K} f(x^k) = f(x^*).$$

Pela definição do Algoritmo 5.1, como $k_{j+1} \geq k_j + 1$, temos:

$$f(x^{k_{j+1}}) \leq f(x^{k_j+1}) \leq f(x^{k_j}) + \alpha t_{k_j} \nabla f(x^{k_j})^T d^{k_j} < f(x^{k_j})$$

para todo $j \in I\!N$.

Logo, pela Condição de Armijo, obtemos:

$$\lim_{j \to \infty} t_{k_j} \nabla f(x^{k_j})^T d^{k_j} = 0.$$

Portanto, pelas condições do Passo 2,

$$\lim_{j \to \infty} t_{k_j} \|\nabla f(x^{k_j})\| \|d^{k_j}\| = 0. \tag{5.1}$$

Portanto, existe $K_1 \underset{\infty}{\subset} K$ tal que pelo menos uma das duas possibilidades seguintes é verificada:

(a) $\lim_{k \in K_1} \nabla f(x^k) = 0$

(b)

$$\lim_{k \in K_1} t_k \|d^k\| = 0. \tag{5.2}$$

No Caso (a), deduzimos que $\nabla f(x^*) = 0$ e a tese está provada.

No Caso (b), existe $K_2 \underset{\infty}{\subset} K_1$ tal que pelo menos uma das duas seguintes possibilidades é verificada:

(c) $\lim_{k \in K_2} \|d_k\| = 0$.

(d) $\lim_{k \in K_2} t_k = 0$.

No caso (c), as condições do Passo 2 também implicam que $\nabla f(x^k) \to 0$ para $k \in K_2$ e, portanto, $\nabla f(x^*) = 0$.

Consideremos o Caso (d). Sem perda de generalidade, suponhamos que $t_k < 1$ para todo $k \in K_2$. Portanto, pelo Passo 3 do algoritmo, para todo $k \in K_2$ existe $\bar{t}_k > 0$ tal que

$$f(x^k + \bar{t}^k d^k) > f(x^k) + \alpha \bar{t}_k \nabla f(x^k)^T d^k \tag{5.3}$$

Mais ainda, como $\bar{t}_k \leq M t_k$ e, por (5.2), $t_k \|d_k\| \to 0$,

$$\lim_{k \in K_2} \bar{t}_k \|d_k\| = 0.$$

Portanto, definindo $s^k = \bar{t}_k d^k$ para todo $k \in K_2$, temos:

$$\lim_{k \in K_2} \|s_k\| = 0. \tag{5.4}$$

Por (5.3) e o Teorema de Valor Médio, para todo $k \in K_2$ existe $\xi_k \in [0,1]$ tal que

$$\nabla f(x^k + \xi^k s^k)^T s^k = f(x^k + s^k) - f(x^k) > \alpha \nabla f(x^k)^T s^k. \tag{5.5}$$

Também, pelo Passo 2,

$$\frac{\nabla f(x^k)^T s^k}{\|s^k\|} \leq -\theta \|\nabla f(x^k)\| \tag{5.6}$$

para todo $k \in K_2$.

Sejam $K_3 \underset{\infty}{\subset} K_2$, $s \in I\!R^n$ tais que $\lim_{k \in K_1} s^k / \|s^k\| = s$.

Por (5.4), dividindo ambos lados da desigualdade (5.5) por $\|s^k\|$, e tomando limites para $k \in K_3$, obtemos:

$$\nabla f(x^*)^T s \geq \alpha \nabla f(x^*)^T s.$$

Como $\alpha < 1$ e $\nabla f(x^k)^T s^k < 0$ para k, isto implica que $\nabla f(x^*)^T s = 0$. Portanto, tomando limites em (5.6), obtemos que $\nabla f(x^*) = 0$. **QED**

5.3 Convergência Local

A Convergência Global no sentido indicado na seção anterior é uma propriedade muito desejável para algoritmos de minimização sem restrições, pois ela garante que convergência a pontos onde o gradiente é diferente de zero não pode acontecer.

Entretanto, a eficiência prática dos algoritmos está (também) ligada a outras propriedades teóricas, de Convergência Local. Estas propriedades dizem que, quando a seqüência gerada pelo algoritmo passa perto de uma solução, tal proximidade é reconhecida e o algoritmo converge rapidamente a ela.

Para que isso aconteça, precisamos que a distância entre x^{k+1} e x^k seja pequena quando $\nabla f(x^k)$ é pequeno. Vamos formalizar esse requisito no seguinte axioma.

Axioma 5.1

O Algoritmo 5.1 é implementado de maneira que existe $b > 0$ tal que

$$\|x^{k+1} - x^k\| \leq b \|\nabla f(x^k)\| \tag{5.7}$$

para todo $k \in I\!N$.

Este axioma é compatível com buscas lineares que obedecem as condições do Passo 3.

Nossa estratégia para provar convergência local superlinear tem três partes. No Teorema 5.3 provamos que, se x^* é um ponto limite isolado, a seqüência inteira converge a ele. No Teorema 5.4 provamos que, se o algoritmo começa perto de um minimizador local estrito, a seqüência também converge. Apesar de análogos, nenhum destes teoremas se deduz do outro. Entretanto, ambos mostram que a convergência de toda a seqüência para um ponto x^* pode ser esperada em muitos casos. Com essa hipótese, e supondo que as direções d^k são obtidas com uma filosofia newtoniana, provaremos convergência superlinear.

Diremos que x^* é um ponto *isolado* se existe $\epsilon > 0$ tal que $\nabla f(x) \neq 0$ para todo $x \in B(x^*, \epsilon) - \{x^*\}$.

Teorema 5.3. *Suponhamos que x^* é isolado, a seqüência $\{x^k\}$ é gerada pelo Algoritmo 5.1 com o Axioma 5.1 e $\lim_{k \in K} x^k = x^*$ para uma subseqüência $K \underset{\infty}{\subset} I\!\!N$. Então, $\nabla f(x^*) = 0$ e*

$$\lim_{k \to \infty} x^k = x^*.$$

Prova. O fato de que $\nabla f(x^*) = 0$ é conseqüência do Teorema 5.2. Pela continuidade de ∇f temos, ao mesmo tempo, que $\lim_{k \in K} \nabla f(x^k) = 0$. Portanto, pelo Axioma 5.1,

$$\lim_{k \in K} \|x^{k+1} - x^k\| = 0.$$

Mas, como x^* é isolado, existe $\epsilon > 0$ tal que $\nabla f(x) \neq 0$ para todo $x \in B(x^*, \epsilon) - \{x^*\}$. Logo, pela hipótese do teorema, existe $k_1 \in K$ tal que

$$\|x^{k+1} - x^k\| < \epsilon/2 \text{ e } \|x^k - x^*\| < \epsilon/2$$

para todo $k \in K, k \geq k_1$.

Definamos

$$C = \{x \in I\!\!R^n \mid \epsilon/2 \leq \|x - x^*\| \leq \epsilon\}.$$

Claramente, C é compacto e não contém pontos onde o gradiente se anula. Portanto, pelo Teorema 5.2, C não contém infinitos iterandos. Em conseqüência, temos duas possibilidades:

1. Existe $k_2 \in I\!\!N$ tal que $\|x^k - x^*\| \leq \epsilon/2$ para todo $k \geq k_2$.

2. Existem infinitos iterandos $k \geq k_1$, tais que $\|x^k - x^*\| \leq \epsilon/2$ e $\|x^{k+1} - x^k\| > \epsilon/2$.

No primeiro caso, como x^* é o único ponto limite possível na bola com raio $\epsilon/2$, temos que a seqüência $\{x^k\}$ converge a x^*.

Vejamos o segundo caso. Seja $K_1 \underset{\infty}{\subset} I\!N$ tal que $\|x^k - x^*\| \leq \epsilon/2$ e $\|x^{k+1} - x^k\| > \epsilon/2$ para todo $k \in K_1$.

Como todos os iterandos $\{x^k\}_{k\in K_1}$ estão na bola com raio $\epsilon/2$ e x^* é o único limite possível nessa bola, temos que

$$\lim_{k\in K_1} x^k = x^*.$$

Portanto,

$$\lim_{k\in K_1} \nabla f(x^k) = 0.$$

Pelo Axioma 5.1, isto implica que

$$\lim_{k\in K_1} \|x^{k+1} - x^k\| = 0,$$

o que contradiz o fato de que $\|x^{k+1} - x^k\| \geq \epsilon/2 \; \forall \; k \in K_1$. Portanto, o segundo caso mencionado acima é impossível e, em conseqüência, a prova está completa. **QED**

Teorema 5.4. *Suponhamos que o ponto isolado x^* é um minimizador local estrito e que a seqüência $\{x^k\}$ é gerada pelo Algoritmo 5.1 com o Axioma 5.1. Então, existe $\delta_1 > 0$ tal que, sempre que $\|x^0 - x^*\| \leq \delta_1$ teremos que*

$$\lim_{k\to\infty} x^k = x^*.$$

Prova. Seja $\epsilon > 0$ tal que x^* é minimizador global estrito de f na bola $B(x^*, \epsilon)$ e suponhamos que esta bola não contém nenhum outro ponto onde o gradiente se anule. Pelo Axioma 5.1, existe $\delta \in (0, \epsilon/2)$ tal que

$$\|x^k - x^*\| \leq \delta \Rightarrow \|x^{k+1} - x^k\| \leq \epsilon/2. \tag{5.8}$$

Seja agora c o mínimo de $f(x)$ no conjunto definido por $\delta \leq \|x - x^*\| \leq \epsilon$. Seja $\delta_1 \in (0, \delta)$ tal que

$$\|x - x^*\| \leq \delta_1 \Rightarrow f(x) < c.$$

Provemos por indução que, tomando $\|x^0 - x^*\| \leq \delta_1$, obtemos $\|x^k - x^*\| \leq \epsilon/2$ e $f(x^k) < c$ para todo k. Pela definição de δ_1 isto é verdade se $k = 0$. Para o passo indutivo, observe que, por (5.8), temos que $\|x^{k+1} - x^*\| \leq \epsilon$. Mas, pela definição de c e o fato de que $f(x^{k+1}) < f(x^k)$, isto implica que $\|x^{k+1} - x^*\| \leq \epsilon/2$.

Portanto, a seqüência completa está contida em $B(x^*, \epsilon/2)$. Como x^* é o único ponto onde o gradiente se anula nesta bola, o Teorema 5.3 implica que a seqüência inteira converge para x^*, como queriamos provar. **QED**

O próximo algoritmo é um caso particular do Algoritmo 5.1 e estabelece a forma em que a direção d^k é calculada. Esta direção será a solução aproximada de um sistema linear $B_k d = -\nabla f(x^k)$, o que aproxima nossa filosofia do paradigma newtoniano. Mais ainda, a opção newtoniana é acentuada pelo fato de que o passo $t_k = 1$ é preferencial: se esse passo satisfaz a condição de Armijo, terminamos nele a busca linear. Por último, o axioma estabelece que as matrizes B_k serão definidas positivas e as normas de suas inversas limitadas.

Algoritmo 5.2. Na implementação do Algoritmo 5.1:

- A direção d^k é solução de
$$B_k d = -\nabla f(x^k) + r^k,$$
onde $B_k \in I\!R^{n\times n}$ é simétrica e definida positiva, e
$$\|r^k\| \leq \eta_k \|\nabla f(x^k)\|$$
com $\eta_k \in [0, 1)$.

- Se
$$f(x^k + d^k) \leq f(x^k) + \alpha \nabla f(x^k)^T d^k,$$
escolhemos $t_k = 1$ e $x^{k+1} = x^k + d^k$.

- O conjunto $\{\|B_k^{-1}\|, k \in I\!N\}$ é limitado.

É fácil ver que, se o conjunto $\{\|B_k\|, k \in I\!N\}$ é limitado, a condição $\|d^k\| \geq \beta\|\nabla f(x^k)\|$ é satisfeita, como requerido pelo Algoritmo 5.1. Mais ainda, se o número de condição $\|B_k\|\|B_k^{-1}\|$ é menor ou igual que $\frac{1}{\theta}$, a condição do ângulo, dada no Passo 2 do algoritmo, também vale. Naturalmente, sempre podem ser encontradas matrizes B_k satisfazendo esses requisitos.

O próximo teorema completa a teoria de convergência básica do Algoritmo 5.1 Mostraremos que, sob o Axioma 5.1, se a seqüência gerada pelo Algoritmo 5.2 converge a um minimizador local onde a Hessiana é definida positiva e as matrizes B_k são aproximações da Hessiana no sentido de Dennis e Moré, então a convergência é superlinear e, para k suficientemente grande, $t_k = 1$. Isto significa que, para k suficientemente grande, precisaremos somente uma avaliação de função por iteração.

Teorema 5.5. *Suponhamos que:*

1. *A seqüência* $\{x^k\}$ *é gerada pelo Algoritmo 5.2 com o Axioma 5.1;*

2. f *admite derivadas segundas contínuas em uma vizinhança de* x^**;*

3. $\nabla^2 f(x^*) > 0$*;*

4. $\lim_{k\to\infty} x^k = x^*$*;*

5. *A condição Dennis-Moré*

$$\lim_{k\to\infty} \frac{\|[B_k - \nabla^2 f(x^k)]d^k\|}{\|d^k\|} = 0$$

e a condição Newton-Inexata

$$\lim_{k\to\infty} \eta_k = 0$$

se verificam.

Então,

- *Existe* $k_0 \in I\!N$ *tal que* $t_k = 1$ *para todo* $k \geq k_0$*.*
- *A seqüência* $\{x^k\}$ *converge superlinearmente a* x^**.*

Prova. Pela fórmula de Taylor, temos:

$$f(x^k+d^k)-f(x^k)-\alpha\nabla f(x^k)^T d^k = (1-\alpha)\nabla f(x^k)^T d^k+\frac{1}{2}(d^k)^T\nabla^2 f(x^k)d^k+o(\|d^k\|^2)$$

$$= (1-\alpha)(d^k)^T[\nabla f(x^k) + \nabla^2 f(x^k)d^k] + \left(\alpha - \frac{1}{2}\right)(d^k)^T\nabla^2 f(x^k)d^k + o(\|d^k\|^2).$$

Mas $B_k d^k + \nabla f(x^k) = r^k$ e, pelo Passo 2 do algoritmo e $\eta_k \to 0$, temos que $\|r^k\| = o(\|\nabla f(x^k)\|) = o(\|d^k\|)$. Portanto,

$$f(x^k + d^k) - f(x^k) - \alpha\nabla f(x^k)^T d^k$$

$$= (1-\alpha)(d^k)^T r^k+(1-\alpha)(d^k)^T[\nabla^2 f(x^k)-B_k]d^k+\left(\alpha-\frac{1}{2}\right)(d^k)^T\nabla^2 f(x^k)d^k+o(\|d^k\|^2)$$

$$= (1-\alpha)(d^k)^T[\nabla^2 f(x^k) - B_k]d^k + \left(\alpha - \frac{1}{2}\right)(d^k)^T\nabla^2 f(x^k)d^k + o(\|d^k\|^2).$$

Mas, pela condição Dennis-Moré:

$$(1-\alpha)(d^k)^T[\nabla^2 f(x^k) - B_k]d^k = o(\|d^k\|^2),$$

portanto:

$$f(x^k + d^k) - f(x^k) - \alpha(d^k)^T \nabla f(x^k) = \left(\alpha - \frac{1}{2}\right)(d^k)^T \nabla^2 f(x^k) d^k + o(\|d^k\|^2). \quad (5.9)$$

Seja $\mu > 0$ um limitante inferior para os autovalores de $\nabla^2 f(x^*)$. Então, existe k_1 tal que $\mu/2$ é um limitante inferior para os autovalores de $\nabla^2 f(x^k)$ para todo $k \geq k_1$. Logo, para todo $k \geq k_1$, temos:

$$\frac{(d^k)^T \nabla^2 f(x^k) d^k}{\|d^k\|^2} \geq \mu/2.$$

Como $\alpha < 1/2$, por (5.9), temos:

$$\frac{f(x^k + d^k) - f(x^k) - \alpha(d^k)^T \nabla f(x^k)}{\|d^k\|^2} \leq \left(\alpha - \frac{1}{2}\right)\frac{\mu}{2} + \frac{o(\|d^k\|^2)}{\|d^k\|^2}.$$

para $k \geq k_2$. Mas, como $\{\|B_k^{-1}\|, k \in I\!N\}$ é limitada e $\nabla f(x^k) \to 0$, pela definição de r^k e $\eta_k \to 0$, temos que $\|d^k\| \to 0$. Portanto, tomando limite no membro da direita da última desigualdade para $k \to \infty$, obtemos:

$$f(x^k + d^k) - f(x^k) - \alpha \nabla f(x^k)^T d^k \leq 0$$

para k suficientemente grande. Logo, pela definição do algoritmo, existe $k_0 \in I\!N$ tal que $t_k = 1$ para $k \geq k_0$. Portanto, a primeira parte da tese está provada.

Pela primeira parte da tese e a definição do Algoritmo 5.2, temos que

$$x^{k+1} - x^k = d^k \text{ para todo } k \geq k_0.$$

Logo, por Taylor,

$$\nabla f(x^{k+1}) = \nabla f(x^k) + \nabla^2 f(x^k) d^k + o(\|d^k\|)$$

$$= B_k d^k + \nabla f(x^k) + [\nabla^2 f(x^k) - B_k] d^k + o(\|d^k\|)$$
$$= r^k + [\nabla^2 f(x^k) - B_k] d^k + o(\|d^k\|).$$

Da mesma maneira que na primeira parte desta prova, temos que $\|r^k\| = o(\|d^k\|)$, portanto:

$$\nabla f(x^{k+1}) = [\nabla^2 f(x^k) - B_k] d^k + o(\|d^k\|).$$

Logo, pela condição Dennis-Moré,

$$\lim_{k \to \infty} \frac{\|\nabla f(x^{k+1})\|}{\|x^{k+1} - x^k\|} = 0.$$

Pelo Axioma 5.1, a continuidade e não-singularidade da Hessiana em x^*, deduzimos que

$$\lim_{k\to\infty} \frac{\|x^{k+1}-x^*\|}{\|x^{k+1}-x^k\|} = 0.$$

Portanto,

$$\lim_{k\to\infty} \frac{\|x^{k+1}-x^*\|}{\|x^{k+1}-x^*\| + \|x^k - x^*\|} = 0.$$

Logo,

$$\lim_{k\to\infty} 1 + \frac{\|x^k - x^*\|}{\|x^{k+1}-x^*\|} = \infty$$

e, em conseqüência,

$$\lim_{k\to\infty} \frac{\|x^{k+1}-x^*\|}{\|x^k - x^*\|} = 0,$$

ou seja, a convergência é superlinear. **QED**

O último teorema de convergência local supõe existência e continuidade das derivadas segundas em x^*. Entretanto, sabemos que a função Lagrangiano Aumentado $L_\rho(x, \lambda, \mu)$ não tem derivadas segundas com respeito a x nos pontos em que $g_i(x) + \mu_i/\rho = 0$. Este fato não é grave, e não anula o poder explicativo do teorema por duas razões. A primeira é que em muitos problemas de PNL vale a propriedade chamada *Complementaridade Estrita*, que diz que, se $g_i(x^*) = 0$, o multiplicador correspondente é maior que zero. Logo, igualdade $g_i(x) + \mu_i/\rho = 0$ seria pouco provável perto de um ponto desse tipo, supondo que μ fosse uma boa aproximação dos multiplicadores. A segunda razão é que, se as funções f, g_i, h tem derivadas segundas contínuas, o gradiente do Lagrangiano Aumentado tem uma propriedade chamada *semisuavidade*, graças a qual a localidade e superlinearidade de métodos Newtonianos é preservada.

É importante interpretar corretamente os resultados desta seção para entender o que efetivamente acontece em situações práticas. O Teorema de Convergência Global é uma espécie de marco geral para o comportamento do algoritmo, mas o leitor usuário de algoritmos computacionais não observará, na prática, iterandos pulando entre *diferentes* pontos de acumulação da seqüência $\{x^k\}$. A razão é dada pelos Teoremas 5.2 e 5.3. Eles afirmam que pontos estacionários isolados são poderosos atratores para o algoritmo. Isto significa que, na prática, o algoritmo gerará uma seqüência que tende a infinito, sem pontos de acumulação, ou convergirá a um ponto x^*. A possibilidade $\|x^k\| \to \infty$ é descartada se os conjuntos de nível de f são limitados. Neste contexto, a hipótese $x^k \to x^*$ do Teorema 5.3 não é arbitrária. Agora, este teorema inclui hipóteses adicionais cuja pertinência deve ser discutida.

A Condição Dennis-Moré é uma dessas hipóteses. Ela afirma que

$$\|[B_k - \nabla^2 f(x^k)]d^k\| = o(\|d^k\|).$$

Esta condição é obviamente satisfeita se escolhermos $B_k = \nabla^2 f(x^k)$ e tende a ser satisfeita se B_k é próxima da Hessiana verdadeira, em particular, se $\|B_k - \nabla^2 f(x^k)\| \to 0$. Entretanto, a Condição Dennis-Moré é satisfeita em condições muitas vezes menos exigentes. Com efeito, observemos primeiro que, por Taylor,

$$\|\nabla^2 f(x^k)d^k - [\nabla f(x^k + d^k) - \nabla f(x^k)]\| = o(\|d^k\|).$$

Portanto, a Condição Dennis-Moré será satisfeita sempre que

$$\|B_k d^k - [\nabla f(x^k + d^k) - \nabla f(x^k)]\| = o(\|d^k\|).$$

Agora suponhamos que escolhemos as sucessivas matrizes B_k de maneira a satisfazer a *Equação Secante*

$$B_{k+1}d^k = \nabla f(x^k + d^k) - \nabla f(x^k),$$

ou a mais freqüente (e muitas vezes equivalente) $B_{k+1}(x^{k+1} - x^k) = \nabla f(x^{k+1}) - \nabla f(x^k)$.

Nesse caso, para que a Condição Dennis-Moré seja satisfeita, será suficiente que $\lim_{k\to\infty} \|B_{k+1} - B_k\| = 0$. Durante algumas décadas, grande parte dos esforços em Otimização Numérica estiveram dirigidos à construção de métodos onde os requisitos acima fossem compatíveis. Tais métodos, chamados *secantes*, evitam o cálculo de derivadas segundas nas iterações do algoritmo irrestrito. Aqui nos bastará observar que a Condição Dennis-Moré é a ferramenta chave do ponto de vista teórico e que ela tende a ser satisfeita quando B_k é parecida com a Hessiana, ou quando $B_k d^k \approx \nabla^2 f(x^k)d^k$.

Outra hipótese importante é $\lim_{k\to\infty} \eta_k = 0$. A interpretação literal desta hipótese exigiria a definição de uma seqüência infinita e convergente a zero, como $\{1/k\}, \{1/k^2\}$, etc. Uma interpretação menos vulgar é que η_k é a tolerância para o erro na resolução do sistema linear $B_k d = -\nabla f(x^k)$, medida por comparação dos resíduos em $d = 0$ e na solução encontrada. Quanto menor for esta tolerância, mais exata é a resolução dos sistemas lineares, e mais podemos esperar algo parecido à convergência superlinear. Na maioria das vezes, haverá duas possibilidades: resolvemos exatamente o sistema linear ou o resolvemos de maneira inexata, por algum método iterativo, geralmente Gradientes Conjugados. No primeiro caso, η_k é muito pequeno, mas não exatamente zero, pois no computador trabalhamos com precisão alta, mas não infinita. De todos modos, a resolução exata do sistema é frequentemente associada à observação de convergência superlinear. Quando se resolve o sistema por um método iterativo, o mais usual é fixar um único valor para η_k, pequeno, aspirando com isso a que a convergência, se não for superlinear, seja pelo menos bastante rápida.

Agora: como é observada na prática a convergência superlinear? Antes de responder a esta pergunta, formulemos a imediatamente anterior: É observável o fato de que $t_k = 1$, ou seja, que cada iteração envolve, assintoticamente, uma única avaliação de função? A resposta é *sim*. Em problemas bem comportados se observa que a primeira tentativa $t = 1$ produz o decréscimo esperado e, em conseqüência, $x^{k+1} = x^k + d^k$, quando B_k é parecida com a Hessiana verdadeira e a resolução do sistema linear é razoavelmente precisa. Se tal coisa não acontece, isso se deve, em geral, a que a Hessiana na solução é quase singular ou muito mal-condicionada.

Por último, a convergência superlinear consiste em que $\|x^{k+1}-x^*\|/\|x^k-x^*\| \to 0$, o que, se $\nabla^2 f(x^*)$ é não-singular, equivale a

$$\lim_{k\to\infty} \frac{\|\nabla f(x^{k+1})\|}{\|\nabla f(x^k)\|} = 0.$$

O que se observa, quando tudo vai bem, é que $\|\nabla f(x^{k+1})\|$ decresce significativamente em relação a $\|\nabla f(x^k)\|$, mas o usuário não deve esperar uma convergência acadêmica e impecável para zero. Ou seja, se observamos que, em sucessivas iterações, a norma do gradiente é, digamos, da ordem da metade da norma do gradiente anterior, isto não significa que a convergência superlinear esteja sendo violada. O limite, afinal de contas, só acontece no infinito.

5.4 Cálculo das Direções de Busca

O Método de Lagrangiano Aumentado é especialmente adequado para problemas de grande porte. Por isso, interessam procedimentos para calcular d^k que envolvam pouca memória e pouco tempo computacional. Via de regra, consideramos proibitivo o armazenamento completo de uma matriz $B_k \in \mathbb{R}^{n\times n}$. A maneira mais radical de poupar tempo e memória por iteração consiste em escolher B_k como um múltiplo da matriz Identidade. Isto é, $B_k = \sigma_k I$, onde σ_k deve ser maior que zero para que a matriz seja definida positiva. Dessa maneira:

$$d^k = -(\sigma_k I)^{-1}\nabla f(x^k) = -\frac{1}{\sigma_k}\nabla f(x^k).$$

Definamos $s^k = x^{k+1} - x^k$, $y^k = \nabla f(x^{k+1}) - \nabla f(x^k)$. A melhor forma de escolher σ_k nesse caso dá origem ao Método de Gradiente Espectral, introduzido por Raydan em 1997. A idéia é satisfazer a equação secante $B_{k+1}s^k = y^k$ da melhor maneira possível dentro das limitação de que as matrizes B_k sejam múltiplos da Identidade. Assim, minimizando $\|\sigma I s^k - y^k\|^2$ em relação a σ, obtemos:

$$\sigma_{k+1} = \frac{(s^k)^T y^k}{(s^k)^T s^k}.$$

Como σ_{k+1}, assim definido, poderia ser nulo, negativo ou excessivamente grande, ele é corrigido com salvaguardas $\sigma_{min}, \sigma_{max} > 0$, ou seja, em definitiva:

$$\sigma_{k+1} \leftarrow \min\{\sigma_{max}, \max\{\sigma_{min}, \sigma_{k+1}\}\}.$$

Desta maneira, as matrizes B_k são definidas positivas e todas as propriedades de convergência global e local, exceto o Teorema 5.5, se verificam.

Com buscas unidimensionais que exijam $f(x^{k+1}) < f(x^k)$ para todo k (ditas *monótonas*), o Método do Gradiente Espectral não funciona muito bem. O desempenho melhora extraordinariamente se, em vez de forçar o decréscimo de f em todas as iterações, se exige apenas que f diminua, digamos, cada 10 passos. Por exemplo, a Condição de Armijo pode ser substituída por:

$$f(x^{k+1}) \leq f(x^k + t_k d^k) \leq \max\{f(x^k), \ldots, f(x^{k-10})\} + \alpha t_k \nabla f(x^k)^T d^k.$$

Na busca unidimensional começamos nossos testes para t com $t = 1$, de maneira de aproveitar as prováveis boas propriedades da escolha espectral.

A memória exigida pelo Método do Gradiente Espectral é realmente mínima, o que faz que para dimensões extremamente grandes seja o único método utilizável. Nos últimos tempos há um revigoramento dos métodos de gradientes conjugados para funções gerais (não quadráticas) mas ainda é cedo para uma avaliação dos progressos neste sentido.

5.4.1 Newton-Truncado

Em `Algencan` há duas formas de calcular as direções de busca: chamadas Newton-Truncado (NT) e Quase-Newton (QN). Ambas são aptas para lidar com problemas de grande porte.

A idéia NT é simples: consideramos o problema quadrático:

$$\text{Minimizar } \frac{1}{2} d^T \nabla^2 f(x^k) d + \nabla f(x^k)^T d$$

e tratamos de encontrar uma solução aproximada d usando o clássico Método de Gradientes Conjugados (GC) de Hestenes e Stiefel. Para isso, começamos as iterações de GC por $d = 0$ e, em cada iteracão GC, testamos as condições do Passo 2 do Algoritmo 5.1. Tomamos providências para ficar com um iterando d que verifique estas condições. Isto não é dificil porque na primeira iteração do Método GC o iterando é um múltiplo de $-\nabla f(x^k)$. Se essas condições se cumprem, as iterações GC continuam até que o problema quadrático é resolvido com uma precisão dada.

Quando a Hessiana $\nabla^2 f(x^k)$ não é definida positiva, o problema de minimização quadrática acima pode não ter solução. Nesse caso, o Método GC se interrompe, mas a possibilidade de calcular uma direção adequada do ponto de vista do Passo 2 do algoritmo, não é afetada. O pior que pode acontecer é que sejamos obrigados a escolher um múltiplo da direção do gradiente.

Em cada passo do Método de Gradientes Conjugados é necessário calcular produtos Hessiana × Vetor. Às vezes, não queremos fornecer a Hessiana, porque não podemos calculá-la ou porque julgamos seu cálculo pouco econômico. Nesse caso, que é o mais freqüente, cada vez que aparece um produto desse tipo, o substituímos por um quociente incremental, usando a aproximação:

$$\nabla^2 f(x^k)d \approx \frac{1}{h}[\nabla f(x^k + hd) - \nabla f(x^k)]$$

com um valor pequeno de h. Nestas circunstâncias, cada iteração do Método de Gradientes Conjugados envolve uma avaliação adicional do gradiente de f.

5.4.2 Quase-Newton e Precondicionadores

Nesta seção abandonamos a simplificação de chamar f à função objetivo dos subproblemas de Lagrangiano Aumentado porque planejamos usar a estrutura específica desta função.

Se trata de criar uma fórmula quase-Newton adequada à forma da Hessiana do Lagrangiano Aumentado. Esta Hessiana está bem definida sempre que $\mu_i + \rho g_i(x) \neq 0$. Por cálculos diretos, constatamos:

$$\nabla^2 L_\rho(x, \lambda, \mu) = \nabla^2 f(x) + A(x) + G(x),$$

onde

$$A(x) = \rho\Big[\sum_{i=1}^{m} \nabla h_i(x) \nabla h_i(x)^T + \sum_{i \in I(x)} \nabla g_i(x) \nabla g_i(x)^T\Big],$$

$$G(x) = \sum_{i=1}^{m} [\lambda_i + \rho h_i(x)] \nabla^2 h_i(x) + \sum_{i \in I(x)} [\mu_i + \rho g_i(x)] \nabla^2 g_i(x),$$

e

$$I(x) = \{i \in \{1, \dots, p\} \mid \mu_i + \rho g_i(x) > 0\}.$$

Suponhamos que x^c é a iteração atual no Algoritmo 5.1 e que x^p é a iteração anterior.

Por simplicidade, escreveremos $L(x) = L_\rho(x, \lambda, \mu)$.

Queremos definir uma aproximação razoável e barata de $\nabla^2 L(x^c)$. Definimos:

$$s = x^c - x^p, \ \ y = \nabla L(x^c) - \nabla L(x^p).$$

Para obter a aproximação da Hessiana, a matriz $A(x)$ será corrigida duas vezes. Na primeira correção, acrescentamos uma matriz diagonal σI, com o objetivo de garantir não-singularidade. (Observe que $A(x)$ é sempre semidefinida positiva.) Com efeito, seguindo a filosofia espectral, definimos:

$$\sigma_{spec} = \text{ Argmin } \|(A(x) + \sigma I)s - y\|^2.$$

Isto implica:

$$\sigma_{spec} = \frac{(y - A(x)s)^T s}{s^T s}.$$

Por segurança, corrigimos:

$$\sigma = \max\{\sigma_{min}, \min\{\sigma_{max}, \sigma_{spec}\}\}$$

e, finalmente,

$$A_+ = A(x) + \sigma I.$$

Se $s^T y \leq 10^{-8}\|s\|\|y\|$, definimos $H = A_+$. Em caso contrário, corrigimos A_+ de maneira a satisfazer a equação secante, mantendo a definição-positiva de H. Como A_+ é definida positiva, o natural é corrigir esta matriz usando uma fórmula famosa chamada BFGS. Assim:

$$H = A_+ + \frac{yy^T}{s^T y} - \frac{A_+ s s^T A_+}{s^T A_+ s}.$$

Com isto, pode-se verificar a equação secante e o fato de que H continua definida positiva.

Isto completa a definição da aproximação da Hessiana.

A direção de busca será obtida resolvendo um sistema linear cuja matriz é H, com o Método GC. Um bom precondicionador para este sistema pode ser definido da seguinte maneira.

1. Defina $D = Diag(A(x))$.

2. Calcule o coeficiente espectral:

$$\sigma_P = \max\left\{\sigma_{min}, \min\left\{\sigma_{max}, \frac{(y - Ds)^T s}{s^T s}\right\}\right\}.$$

3. Calcule $D_+ = D + \sigma_P I$.

4. Como no cálculo de H, se $s^T y \leq 10^{-8}\|s\|\|y\|$, definimos $H_P = D_+$. Se não, definimos

$$H_P = D_+ + \frac{yy^T}{s^T y} - \frac{D_+ s s^T D_+}{s^T D_+ s}.$$

(Observe que H_P não precisa ser calculado explicitamente.)

Neste processo, H_P é a correção BFGS de uma matriz definida positiva e, portanto, resulta também definida positiva. Sua inversa vem dada por:

$$H_P^{-1} = D_+^{-1} + \frac{(s - D_+^{-1}y)s^T + s(s - D_+^{-1}y)^T}{s^T y} - \frac{(s - D_+^{-1}y)^T y s s^T}{(s^T y)^2}.$$

A explicitude desta fórmula mostra que H_P pode ser usado como precondicionador do Método GC. Mais ainda, nada impede que este precondicionador seja usado também para as iterações GC no esquema Newton-Truncado, o que de fato é feito como primeira opção em `Algencan`.

EXERCÍCIOS

1. Mostre, com um exemplo adequado, a região onde a função Lagrangiano Aumentado não tem derivadas segundas contínuas.

2. Faça desenhos ilustrando a Condição de Armijo.

3. Mostre que, se no Algoritmo 5.1 eliminamos a condição que segue à Condição de Armijo, então passos t_k arbitrariamente e desnecessariamente pequenos poderiam ser tomados, e o algoritmo poderia convergir a um ponto onde o gradiente não se anula.

4. Mostre que o ponto x^{k+1} escolhido no Passo 2 do Algoritmo 5.1 poderia não satisfazer a Condição de Armijo.

5. Defina explicitamente uma seqüência $\{z^k\}$, densa em $I\!R^n$. Modifique o Algoritmo 5.1 da seguinte maneira. Em primeiro lugar, elimine o Passo 1. Em segundo lugar, depois de obtido t_k de acordo com o especificado no Passo 3, calcule $f(z^k)$ e defina $x^{k+1} = x^k + t_k d^k$ se $f(x^k + t_k d^k) \leq f(z^k)$ e $x^{k+1} = z^k$ em caso contrário. Prove que, para este algoritmo, temos que $f(x^k) \to -\infty$ ou todo ponto limite do algoritmo é um minimizador global. Compare este resultado com os resultados obtidos neste Capítulo para o Algoritmo 5.1. Detecte, critique e resolva o aparente paradoxo.

6. Invente métodos de busca linear que se ajustem aos Passos 2 e 3 do Algoritmo 5.1.

7. Complete o argumento do Teorema 5.5, pelo qual

$$\lim_{k\to\infty} \frac{\|\nabla f(x^{k+1})\|}{\|x^{k+1} - x^k\|} = 0 \Rightarrow \lim_{k\to\infty} \frac{\|x^{k+1} - x^*\|}{\|x^{k+1} - x^k\|} = 0.$$

8. Com base nos resultados teóricos deste capítulo, tire conclusões práticas sobre o comportamento do algoritmo `Gencan`. Rode diversos exemplos de sua invenção com este algoritmo e verifique a correção ou incorreção de suas conjecturas. Nos casos negativos, volte sobre os resultados e sua interpretação.

9. Invente problemas de minimização sem restrições com derivadas primeiras contínuas mas com descontinuidade das derivadas segundas do tipo das padecidas pelo Lagrangiano Aumentado. Faça com que as soluções desses problemas sejam pontos onde as derivadas segundas não existem. Estude empiricamente o comportamento de `Gencan` nesses problemas tire conclusões e faça conjecturas.

10. Complete a seção sobre cálculo das direções de busca com as provas insinuadas no texto.

Capítulo 6

Subproblemas com Restrições Simples

No Método de Lagrangiano Aumentado procuramos minimizar $f(x)$ sujeita a $h(x) = 0$, $g(x) \leq 0$ e $x \in \Omega$. Em cada iteração do método, as restrições $h(x) = 0$ e $g(x) \leq 0$ passam à função objetivo e o processo consiste em minimizar a função resultante restrita a $x \in \Omega$. Este procedimento supõe que contamos com algoritmos sensatos para minimizar funções no conjunto Ω. Em princípio, não há restrições para a forma que o conjunto Ω pode adotar, mas o caso mais comum é que se trate de uma *caixa*:

$$\Omega = \{x \in I\!R^n \mid a^{min} \leq x \leq a^{max}\}.$$

Os vetores a^{min} e a^{max}, $a^{min} < a^{max}$ são os limitantes inferiores e superiores da caixa Ω, respectivamente.

Seguindo a simplificação de notação adotada no Capítulo 5, chamaremos $f(x)$ à função objetivo do problema de minimização em caixas. O caso de que a caixa não possua algum limitante inferior ou superior não será tratado explicitamente aqui porque, na prática, o conveniente é adotar limitantes artificiais, de módulos muito grandes, quando os limitantes naturais não existem.

É útil considerar Ω como a união de faces disjuntas. Dado um conjunto de índices $I \subset \{1, \ldots, 2n\}$ chamamos F_I ao conjunto dos pontos $x \in \Omega$ tais que:

1. $x_i = a_i^{min}$ se $i \in I$;

2. $x_i = a_i^{max}$ se $n + i \in I$;

3. $a_i^{min} < x_i < a_i^{max}$ se $i \notin I$ e $n + i \notin I$.

Por exemplo, $F_\emptyset$ é o interior de Ω, $F_{\{1,\ldots,n\}}$ é o conjunto cujo único elemento é $(a_1^{min}, \ldots, a_n^{min})$ e assim por diante. Claramente, se $I \neq J$ temos que $F_I \cap F_J = \emptyset$ e $\cup F_I = \Omega$. Uma face também se caracteriza por suas variaveis livres e suas variáveis fixas. As variáveis livres na face F_I são aquelas variáveis x_i tais que $i \notin I$ e $n+i \notin I$.

As variáveis x_i tais que $i \in I$ se dizem *fixas no extremo inferior* e as variáveis x_i tais que $n + i \in I$ se dizem *fixas no extremo superior*.

Denotamos $\bar{F}_I$ o fecho topológico de F_I. Por exemplo, $\bar{F}_\emptyset = \Omega$.

Os algoritmos mais razoáveis para minimizar em caixas geram pontos x^k que pertencem à caixa para todo k. Portanto, cada ponto x^k pertence a uma das faces de Ω, e somente a uma. A maioria dos algoritmos eficientes se baseiam no *Princípio das Restrições Ativas*, enunciado em continuação.

6.1 Princípio Forte das Restrições Ativas

Suponhamos que nosso objetivo é encontrar um minimizador global da função contínua $f(x)$ sujeita a $x \in \Omega$. Seja $x^0 \in F_{I(x^0)} \subset \Omega$. Suponhamos que a seqüência $\{x^k\}$, gerada a partir do ponto inicial x^0, obedece os seguintes axiomas:

1. Se x^k é minimizador global de f em Ω, a seqüência *pára* em x^k. Em caso contrário, o ponto x^{k+1} satisfaz $f(x^{k+1}) < f(x^k)$.

2. Se x^k não é minimizador global de f em $F_{I(x^k)}$, o ponto x^{k+1} é minimizador global de f em $F_{I(x^k)}$ ou pertence à fronteira $\bar{F}_{I(x^k)} - F_{I(x^k)}$.

As iterações de um algoritmo que obedece os axiomas acima são, em conseqüência, de três tipos diferentes:

- Iterações internas: chamamos assim às iterações nas quais x^k não é minimizador global de f em $F_{I(x^k)}$ e $x^{k+1} \in F_{I(x^k)}$ é minimizador global de f nessa face.

- Iterações fronteira: x^k não é minimizador global de f em $F_{I(x^k)}$ e $x^{k+1} \in \bar{F}_{I(x^k)} - F_{I(x^k)}$.

- Iterações de abandono: x^k é minimizador global de f em $F_{I(x^k)}$ mas, não sendo minimizador global de f em Ω, $x^{k+1} \notin \bar{F}_{I(x^k)}$.

A filosofia que está por trás deste princípio é que uma face deve ser explorada até que um minimizador nela seja atingido ou até que seja atingido um ponto de sua fronteira, o que implica uma mudança de face.

Teorema 6.1. *Suponhamos que a seqüência $\{x^k\}$ obedece os axiomas desta seção. Então, a seqüência é finita, ou seja, existe k tal que x^k é minimizador global de f sujeita a $x \in \Omega$.*

Prova. Suponhamos que a seqüência $\{x^k\}$ é infinita. Como o número de faces é finito e $f(x^k) > f(x^{k+1})$ sempre que x^k não seja o minimizador global, tanto o número de iterações internas como o número de iterações de abandono deve ser finito. Isto significa que existe k_0 tal que para todo $k \geq k_0$ todas as iterações são do tipo fronteira. Mas em cada iteração do tipo fronteira o número de variáveis livres se reduz em pelo menos um. Portanto, também não pode haver infinitas iterações deste tipo. Logo, a seqüência não pode ser infinita. **QED**

6.2 Princípio Prático de Restrições Ativas

O Princípio Forte de Restrições Ativas assinala uma direção algorítmica, mas não gera métodos implementáveis. A razão é que quase nunca podemos pretender atingir um minimizador global de f na face de x^k, muito menos em tempo finito.

Entretanto, o princípio de permanecer em uma mesma face enquanto essa permanência seja promissora é válido. A maneira sensata de permanecer na face à qual pertence o iterando x^k é pensar no problema irrestrito cujas variáveis são as variáveis livres nessa face e aplicar um algoritmo de minimização sem restrições a esse problema. O algoritmo irrestrito terá, provavelmente, a tendência de convergir a um ponto estacionário para as variáveis livres (ou seja, um ponto onde se anulam as derivadas em relação a essas variáveis), mas poderá trombar em uma parede, isto é, poderá se deter em uma face de dimensão menor. Por outro lado, em cada passo será necessário decidir se ainda vale a pena continuar nessa face ou se é desejável abandoná-la para tentar a sorte em uma face de dimensão maior.

O Gradiente Projetado Contínuo

A partir deste momento, neste capítulo, suporemos que a função f tem derivadas contínuas.

A decisão de continuar na mesma face ou abandoná-la pode ser tomada usando as componentes do chamado Gradiente Projetado Contínuo.

Consideremos, dado $x^k \in \Omega$, o problema

$$\text{Minimizar } \nabla f(x^k)^T(x - x^k) + \frac{1}{2}\|x - x^k\|^2 \quad \text{sujeita a} \quad x \in \Omega. \tag{6.1}$$

Completando quadrados na função objetivo, vemos que este problema é equivalente a

$$\text{Minimizar } \|x^k - \nabla f(x^k) - x\|^2 \quad \text{sujeita a} \quad x \in \Omega.$$

A solução $\bar{x}$ deste problema é a projeção de $x^k - \nabla f(x^k)$ em Ω. Por inspeção direta, vemos que esta solução é dada por:

$$\bar{x}_i = \min\{a_i^{max}, \max\{a_i^{min}, x_i^k - \frac{\partial f}{\partial x_i}(x^k)\}\}.$$

Denotaremos $\bar{x} = P_\Omega(x^k - \nabla f(x^k))$. É fácil ver, por outra parte, que $\bar{x}$ é o único ponto estacionário (KKT) do problema (6.1) e que $\bar{x}$ depende continuamente de x^k.

Por outro lado, se x^k fosse um ponto KKT do problema de minimizar $f(x)$ em Ω, também seria um ponto KKT de (6.1), e vice-versa. E esse fato seria equivalente a dizer que $\|\bar{x} - x^k\| = 0$. Todas estas considerações nos levam a definir o gradiente projetado contínuo $g_P(x^k)$ como

$$g_P(x^k) = P_\Omega(x^k - \nabla f(x^k)) - x^k$$

e a definir o grau de estacionariedade de x^k como a norma deste vetor.

Por sua vez, $g_P(x^k)$ pode ser decomposto de maneira única como

$$g_P(x^k) = g_I(x^k) + [g_P(x^k) - g_I(x^k)],$$

onde $g_I(x^k)$ está no subespaço associado a $F_{I(x^k)}$ e $g_P(x^k) - g_I(x^k)$ está no complemento ortogonal. Em outras palavras, $g_I(x^k)$ é igual a $g_P(x^k)$ para as variáveis livres e tem zeros nas variáveis fixas. Obviamente, $g_I(x^k)$ também depende continuamente de x^k. Este vetor se chama Gradiente Interno.

Se $g_I(x^k) = 0$ temos que x^k é um ponto estacionário de $f(x)$ restrita a $x \in F_{I(x^k)}$. Neste caso, nada mais cabe esperar de um algoritmo irrestrito que usa gradientes e a recomendação sensata (sempre que $g_P(x^k) \neq 0$) seria *abandonar a face*. A mesma recomendação deveria ser feita se a norma de $g_I(x^k)$ for pequena em relação à norma de $g_P(x^k)$. Resumindo, em um algoritmo prático, a face $F_I(x^k)$ deveria ser abandonada quando

$$\|g_I(x^k)\| \leq \eta \|g_P(x^k)\|,$$

onde $\eta \in (0, 1)$ seria um parâmetro algorítmico. Em `Gencan` o teste padrão é:

$$\|g_I(x^k)\| \leq 0.1 \|g_P(x^k)\|.$$

6.2.1 Esquema básico de `Gencan`

`Gencan` é o algoritmo para minimizar em caixas adotado no projeto `Tango`. Este método usa o princípio prático de restrições ativas descrito acima. Para minimizar dentro das faces, `Gencan` emprega um algoritmo do tipo Newton Truncado ou um algoritmo Quase Newton. Quando o teste padrão indica que a face deve ser abandonada `Gencan` emprega uma iteração do método de Gradiente Projetado Espectral (SPG).

Iteração SPG

Sejam σ_{min} e σ_{max} números positivos pequeno e grande, respectivamente, definidos independentemente de k. Seja $\alpha \in (0, \frac{1}{2})$. Se $k = 0$, escolhemos $\sigma_k \in [\sigma_{min}, \sigma_{max}]$

arbitrariamente. Se $k > 0$, definimos $y = \nabla f(x^k) - \nabla f(x^{k-1})$ e $s = x^k - x^{k-1}$. Neste caso, o coeficiente espectral σ_k se define:

$$\sigma_k = \max\left\{\sigma_{min}, \min\left\{\sigma_{max}, \frac{s^T y}{s^T s}\right\}\right\}.$$

A direção SPG é um gradiente projetado *escalado* por σ_k:

$$d^k = P_\Omega\Big(x^k - \frac{1}{\sigma_k}\nabla f(x^k)\Big) - x^k.$$

Uma vez calculada a direção SPG, o seguinte iterando é obtido pelos seguintes passos:

(1) $t \leftarrow 1$;

(2) Teste da condição de decréscimo suficiente de Armijo:

$$f(x^k + td^k) \leq f(x^k) + \alpha t \nabla f(x^k)^T d^k.$$

Se esta condição se verifica, defina $t_k = t$ e $x^{k+1} = x^k + t_k d^k$. Em caso contrário, escolha um novo t no intervalo $[0.1t, 0.9t]$ e repita o teste de Armijo.

Lemma 6.1. *A iteração SPG está bem definida.*

Prova. Suponhamos que $g_P(x^k) \neq 0$. Portanto, x^k não é minimizador de $\|x - (x^k - \nabla f(x^k)\|^2$ sujeita a $x \in \Omega$. Logo, x^k não satisfaz a condição de otimalidade deste problema. Mas a condição de otimalidade deste problema é a mesma que a do problema

$$\text{Minimizar } \|x - (x^k - \frac{1}{\sigma_k}\nabla f(x^k)\|^2 \text{ sujeita a } x \in \Omega.$$

Logo, x^k não satisfaz a condição de otimalidade deste problema. Portanto, $x^k \neq P_\Omega(x^k - \frac{1}{\sigma_k}\nabla f(x^k)$. Mais ainda, a funcao objetivo em $P_\Omega(x^k - \frac{1}{\sigma_k}\nabla f(x^k)$ vale menos que em x^k. Como a função objetivo é uma quadrática convexa, a função objetivo ao logo do segmento $[x^k, x^k + d^k]$ é uma parábola convexa. Portanto, a direção d^k deve ser uma direção de descida para a quadrática. Como os gradientes em x^k são iguais, d^k é uma direção de descida para f.

O fato de que, para t suficientemente pequeno, se obtenha a condição de Armijo segue como no Teorema 5.1. Com isso, fica provado que a iteração SPG está bem definida. **QED**

Lemma 6.2. *Suponhamos que existe um conjunto infinito de índices $K = (k_1, k_2, \ldots)$ tal que, em* `Gencan`*, x^{k+1} é obtido por uma iteração SPG para todo $k \in K$. Então,* *todo ponto limite de* $\{x^k\}_{k\in K}$ *é estacionário.*

`Prova.` Pela Condição de Armijo, e o mesmo raciocínio do Teorema 5.2, deduzimos que

$$\lim_{k\in K} t_k \nabla f(x^k)^T d^k = 0.$$

Pela definição da direção SPG, temos:

$$0 > \frac{\sigma_k}{2}\|d^k\|^2 + \nabla f(x^k)^T d^k \geq \nabla f(x^k)^T d^k.$$

Portanto,

$$\lim_{k\in K} t_k \left[\frac{\sigma_k}{2}\|d^k\|^2 + \nabla f(x^k)^T d^k\right] = 0.$$

Consideremos primeiro o caso em que

$$\lim_{k\in K} \frac{\sigma_k}{2}\|d^k\|^2 + \nabla f(x^k)^T d^k = 0.$$

Seja $K_1 \underset{\infty}{\subset} K$ tal que $\lim_{k\in K_1} x^k = x^*$.

Como $\sigma_k \in [\sigma_{min}, \sigma_{max}]$ para todo $k \in K$, existe um subconjunto infinito de índices $K_2 \underset{\infty}{\subset} K_1$ tal que

$$\lim_{k\in K_2} \sigma_k = \sigma > 0.$$

Definimos $Q(d) = \frac{\sigma}{2}\|d\|^2 + \nabla f(x^*)^T d$. Suponhamos que existe $\hat{d} \in I\!R^n$ tal que $x^* + \hat{d} \in \Omega$ e

$$Q(\hat{d}) < 0.$$

Seja $\hat{d}^k = x^* + \hat{d} - x^k$. Claramente, $x^k + \hat{d}^k = x^* + \hat{d} \in \Omega$ para todo $k \in K_2$. Por continuidade, temos:

$$\lim_{k\in K_2} \frac{\sigma_k}{2}\|\hat{d}^k\|^2 + \nabla f(x^k)^T \hat{d}^k = Q(\hat{d}) < 0.$$

Mas, pela definição de d^k:

$$\frac{\sigma_k}{2}\|d^k\|^2 + \nabla f(x^k)^T d^k \leq \frac{\sigma_k}{2}\|\hat{d}^k\|^2 + \nabla f(x^k)^T \hat{d}^k.$$

Portanto, para $k \in K_2$ suficientemente grande,

$$\frac{\sigma_k}{2}\|d^k\|^2 + \nabla f(x^k)^T d^k \leq Q(\hat{d})/2 < 0.$$

Isto contradiz que $\frac{\sigma_k}{2}\|d^k\|^2 + \nabla f(x^k)^T d^k \to 0$. A contradição veio de supor que $\hat{d}$ com a propriedade $Q(\hat{d}) < 0$ poderia existir.

Portanto, para todo $d \in I\!R^n$ tal que $x^* + d \in \Omega$, temos que $Q(d) \geq 0$. Logo, x^* é estacionário.

Passemos agora ao caso em que $\frac{1}{2}\|d^k\|^2 + \frac{1}{\sigma_k}\nabla f(x^k)^T d^k$ não tende a zero.

Portanto, existe $K_3 \underset{\infty}{\subset} K$ tal que

$$\frac{1}{2}\|d^k\|^2 + \frac{1}{\sigma_k}\nabla f(x^k)^T d^k < c < 0,$$

para todo $k \in K_3$, $t_k < 1$,

$$\lim_{k \in K_3} t_k = 0$$

e

$$\lim_{k \in K_3} \sigma_k = \sigma.$$

Pelo algoritmo de escolha de t_k, isto implica que, para todo $k \in K_3$ existe $t'_k \leq 10 t_k$ tal que a Condição de Armijo falha para t'_k.

Logo, para todo $k \in K_3$,

$$f(x^k + t' d^k) > f(x^k) + \alpha t' \nabla f(x^k)^T d^k.$$

Portanto

$$\frac{f(x^k + t' d^k) - f(x^k)}{t'_k} > \alpha \nabla f(x^k)^T d^k.$$

Pelo Teorema de Valor Médio, para todo $k \in K_3$ existe $\xi_k \in [0,1]$ tal que

$$\nabla f(x^k + \xi_k t'_k d^k)^T d^k > \alpha \nabla f(x^k)^T d^k.$$

Como o conjunto de vetores $\{d^k\}$ é limitado, existe um subconjunto infinito de índices $K_4 \underset{\infty}{\subset} K_3$ tal que $\lim_{k \in K_4} d^k = d$.

Tomando limites para $k \in K_4$ nos dois lados da desigualdade, obtemos:

$$\nabla f(x^*)^T d \geq \alpha \nabla f(x^*)^T d.$$

Como $\nabla f(x^k)^T d^k < 0$ para todo k, isto implica que $\nabla f(x^*)^T d = 0$. Portanto,

$$\frac{\sigma}{2}\|d\|^2 + \nabla f(x^*)^T d \geq 0.$$

Logo,

$$\lim_{k \in K_4} \frac{\sigma_k}{2}\|d^k\|^2 + \nabla f(x^k)^T d^k \geq 0.$$

Isto contradiz o fato de que $\frac{\sigma_k}{2}\|d^k\|^2 + \nabla f(x^k)^T d^k \leq c < 0$ para todo $k \in K_4$. **QED**

Axiomas para Iterações na Face

`Gencan` minimiza funções em uma caixa combinando iterações que não modificam face à qual o iterando pertence com iterações SPG, que abandonam a face, quando o teste correspondente indica que isto deve ser feito.

As iterações dentro da face modificam somente as variáveis livres, portanto podem ser consideradas iterações de um algoritmo irrestrito, onde a função objetivo é f, mas sem permitir modificação das variáveis fixas. Iterações consecutivas deste algoritmo devem conduzir a um ponto no qual o gradiente interno se anula ou devem parar de maneira abrupta na fronteira da face. Esta propriedade pode ser formulada nos seguinte axioma:

Axioma Interno

Se $x^k, x^{k+1}, \ldots \in F_I$ é uma seqüência de infinitas iterações obtidas pelo Algoritmo irrestrito interno, então

$$\lim_{j\to\infty} \frac{\partial f}{\partial x_j}(x^k) = 0$$

para todas as variáveis livres x_j. (Isto implica que $g_I(x^{k+j}) \to 0$.)

Os resultados do Capítulo 5 indicam que o Axioma Interno é fácil de verificar se, como algoritmo interno se usa qualquer método irrestrito razoável. Esta afirmação não está isenta de sutileza, pela existência de limitantes para as variáveis livres. Ou seja, um algoritmo irrestrito pode indicar como iterando seguinte um ponto que não verifica as restrições das variáveis livres. Neste caso, o algoritmo deve ter a capacidade de rejeitar esse iterando voltando ao interior da face, ou de bater na fronteira diminuíndo o valor da função.

Convergência de `Gencan`

A descrição algorítmica de `Gencan` pode ser resumida da seguinte maneira:

Algoritmo 6.1.

Seja $\eta \in (0,1)$ o parâmetro algorítmico que fornece o critério para abandonar as faces. Seja $x^0 \in \Omega$ o ponto inicial. Se $x^k \in \Omega$ é um iterando, os passos para obter $x^{k+1} \in \Omega$ ou interromper a execução do algoritmo são os seguintes.

Passo 1. Se $g_P(x^k) = 0$, parar. (O ponto x^k é estacionário do problema de minimizar f em Ω.)

Passo 2. Se $\|g_I(x^k)\| \leq \eta \|g_P(x^k)\|$, obter x^{k+1} usando uma iteração SPG e voltar ao Passo 1.

Passo 3. Se o teste de abandono do Passo 2 não se verifica, obter x^{k+1} tal que $f(x^{k+1}) < f(x^k)$ pelo algoritmo irrestrito interno (logo $x^{k+1} \in F_I$) ou de maneira que x^{k+1} esteja na fronteira de F_I.

Teorema 6.1. *Suponhamos que $\{x^k\}$ é obtida pelo Algoritmo 6.1. Então, a seqüência pára em um ponto estacionário x^k ou admite um ponto de acumulação estacionário.*

Prova. Suponhamos que a seqüência não pára no Passo 1 para nenhum k. Se infinitas iterações são do tipo SPG, a tese segue do Lema 6.1.

Se apenas um número finito de iterações são do tipo SPG, então, existe uma face F_I tal que a partir de certo k_0 todos os iterandos pertencem a F_I. Pelo Axioma Interno, isto implica que $g_I(x^k) \to 0$. Mas, pelo Passo 2, $\|g_I(x^k)\| > \eta \|g_P(x^k)$, portanto $g_P(x^k) \to 0$. Pela continuidade de g_P isto implica que em todo ponto de acumulação g_P se anula. **QED**

O Teorema 6.1 é interessante porque mostra que, em última análise, `Gencan` encontra pontos estacionários. Entretanto, resulta um pouco decepcionante a importância que as iterações SPG têm na prova. Seria mais nobre que o número de iterações SPG fosse finito, de maneira que as propriedades últimas de convergência fossem as do algoritmo irrestrito. No Teorema 6.2 mostramos uma condição suficiente para essa favorável circunstância. Diremos que um ponto estacionário x^* é *dual-degenerado* se existe $i \in \{1, \ldots, n\}$ tal que $i \in I(x^*)$ ou $n + i \in I(x^*)$, mas

$$\frac{\partial f}{\partial x_i}(x^*) = 0.$$

Em outras palavras, em pontos *dual-não-degenerados* somente podem se anular as derivadas parciais correspondentes a variáveis livres.

Teorema 6.2. *Suponhamos que todos os pontos estacionários do problema são dual-não-degenerados. Então o número de iterações SPG é finito.*

Prova. Seja K o conjunto dos índices tais que x^{k+1} é obtido por SPG para todo $k \in K$. Pelo Lema 6.1 existe $K_1 \underset{\infty}{\subset} K$ tal que

$$\lim_{k \in K_1} x^k = x^*$$

e x^* é estacionário. Sem perda de generalidade, podemos supor que existe $i \in \{1, \ldots, n\}$ tal que $x_i^k = a_i^{min}$ mas $x_i^{k+1} > a_i^{min}$ para um conjunto infinito de índices $K_2 \underset{\infty}{\subset} K_1$. Por continuidade, $x_i^* = a_i^{min}$. Entretanto, como x^* é dual-não-degenerado,

$$\frac{\partial f}{\partial x_i}(x^*) > 0.$$

Por continuidade, deduzimos que

$$\frac{\partial f}{\partial x_i}(x^k) > 0$$

para todo $k \in K_2$ suficientemente grande. Portanto, para esses k,

$$g_P(x^k)_i = 0.$$

Isto implica que a restrição $x = a_i^{min}$ não poderia ser abandonada na iteração k. **QED**

EXERCÍCIOS

1. Prove a continuidade de g_P e g_I.

2. Relacione o gradiente interno com o gradiente (com sentido contrário) da função f considerada apenas como dependente das variáveis livres na face $F_{I(x^k)}$.

3. Generalize o Princípio Forte das Restrições Ativas para restrições lineares gerais, de igualdade e desigualdade.

4. Interprete geometricamente a iteração SPG.

5. Mostre que a direção SPG "corrige" o gradiente projetado no sentido de respeitar a dimensionalidade das quantidades envolvidas em seu cálculo. Compare esta propriedade com as propriedades do método de Newton para minimizar sem restrições. Estude o comportamento da direção SPG diante de mudanças de escala em f ou em x.

6. Podemos definir um método onde todas as iterações sejam SPG. Tal método existe e resulta bastante útil no caso em que Ω é um convexo arbitrário onde é fácil projetar. A eficiência do método SPG, entretanto, está vinculada a uma modificação simples, que faz com que não seja necessário diminuír f em todas as iterações. Nesse caso, a condição de Armijo é substituída por

   $$f(x^k + td^k) \leq \max\{f(x^k), \ldots, f(x^{k-10}\} + \alpha t \nabla f(x^k)^T d^k.$$

   (Definimos $f(x^k)$ arbitrário se $k < 0$). Discuta a provável eficiência de SPG em comparação a `Gencan` quando Ω é uma caixa. Prove a convergência a pontos estacionários de SPG.

7. A "escolha de um novo t" na iteração SPG pode ser feita usando interpolações parabólicas com as salvaguardas definidas no algoritmo. Desenvolva as fórmulas adequadas para essas interpolações.

8. Prove que o Axioma Interno implica $g_I(x^{k+j}) \to 0$

9. Interprete geometricamente a degeneração dual.

10. Tire conclusões práticas sobre os teoremas sobre `Gencan` apresentados neste capítulo. Baseados neles, que deveria ser observado quando rodarmos `Gencan` em problemas particulares?

Capítulo 7

Convergência Global

Como no primeiro capítulo, definimos o Problema de Programação Não-Linear:

$$\text{Minimizar } f(x)$$

sujeita a $h(x) = 0, g(x) \leq 0, x \in \Omega$, onde $h : I\!R^n \to I\!R^m$, $g : I\!R^n \to I\!R^p$ e $\Omega \subset I\!R^n$ é, em geral, simples.

Aqui daremos uma versão do Método de Lagrangiano Aumentado onde os subproblemas de minimizar $L_\rho(x, \lambda, \mu)$ não são resolvidos exatamente, mas com uma aproximação que procuraremos definir com cuidado. Para isso, precisaremos especificar um pouco mais a forma do conjunto simples Ω. Seguindo [1] definimos:

$$\Omega = \{x \in I\!R^n \mid \underline{h}(x) = 0, \underline{g}(x) \leq 0\},$$

onde

$$\underline{h} : I\!R^n \to I\!R^{\underline{m}}, \underline{g} : I\!R^n \to I\!R^{\underline{p}}$$

e todas as funções envolvidas têm derivadas primeiras contínuas em um subconjunto suficientemente grande de $I\!R^n$. O operador ∇ denotará sempre derivadas em relação a x.

Quando Ω é uma caixa, temos:

$$\underline{m} = 0, \underline{p} = 2n,$$

$$\underline{g}(x) = (a_1^{min} - x_1, \ldots, a_n^{min} - x_n, x_1 - a_1^{max}, \ldots, x_n - a_n^{max})^T.$$

Quando $\Omega = I\!R^n$ temos $\underline{m} = \underline{p} = 0$.

Dado $\rho > 0, \lambda \in I\!R^m, \mu \in I\!R^{\underline{p}}_+$, o Lagrangiano Aumentado se define, como no Capítulo 1,

$$L_\rho(x, \lambda, \mu) = f(x) + \frac{\rho}{2}\Big\{ \sum_{i=1}^{m} \Big[h_i(x) + \frac{\lambda_i}{\rho}\Big]^2 + \sum_{i=1}^{p} \Big[\max\Big(0, g_i(x) + \frac{\mu_i}{\rho}\Big)\Big]^2 \Big\}. \quad (7.1)$$

O algoritmo principal vem dado em continuação.

Algoritmo 7.1.
Seja $x^0 \in I\!R^n$ um ponto inicial arbitrário.
Os parâmetros para a execução do algoritmo são:

$$\tau \in [0,1), \gamma > 1,$$

$$-\infty < \lambda_{min} < \lambda_{max} < \infty,$$

$$\mu_{max} > 0,$$

$$\rho_1 > 0,$$

$$\bar{\lambda}_i^1 \in [\lambda_{min}, \lambda_{max}] \ \forall i = 1, \ldots, m,$$

$$\mu_i^1 \in [0, \mu_{max}] \ \forall i = 1, \ldots, p.$$

Finalmente, $\{\varepsilon_k\} \subset I\!R_+$ será uma seqüência de parâmetros de tolerância tal que

$$\lim_{k \to \infty} \varepsilon_k = 0. \tag{7.2}$$

Passo 1. *Inicialização*
$k \leftarrow 1$. Definir $V^0 = g(x^0)_+$.

Passo 2. *Resolução do subproblema*
Calcular (se for possível) $x^k \in I\!R^n$ tal que existam $v^k \in I\!R^{\underline{m}}, u^k \in I\!R^{\underline{p}}$ satisfazendo

$$\|\nabla L_{\rho_k}(x^k, \bar{\lambda}^k, \bar{\mu}^k) + \sum_{i=1}^{\underline{m}} v_i^k \nabla \underline{h}_i(x^k) + \sum_{i=1}^{\underline{p}} u_i^k \nabla \underline{g}_i(x^k)\| \leq \varepsilon_k, \tag{7.3}$$

$$u_i^k \geq 0, \underline{g}_i(x^k) \leq \varepsilon_k \text{ para todo } i = 1, \ldots, \underline{p}, \tag{7.4}$$

$$\underline{g}_i(x^k) < -\varepsilon_k \Rightarrow u_i^k = 0 \text{ para todo } i = 1, \ldots, \underline{p}, \tag{7.5}$$

$$\|\underline{h}(x^k)\| \leq \varepsilon_k. \tag{7.6}$$

Passo 3. *Estimar multiplicadores*
Para todo $i = 1, \ldots, m$, calcular

$$\lambda_i^{k+1} = \bar{\lambda}_i^k + \rho_k h_i(x^k) \tag{7.7}$$

e

$$\bar{\lambda}_i^{k+1} \in [\lambda_{min}, \lambda_{max}]. \tag{7.8}$$

Para todo $i = 1, \ldots, p$, calcular

$$\mu_i^{k+1} = \max\{0, \bar{\mu}_i^k + \rho_k g_i(x^k)\}, \tag{7.9}$$

$$V_i^k = \max\left\{g_i(x^k), -\frac{\bar{\mu}_i^k}{\rho_k}\right\},$$

e

$$\bar{\mu}_i^{k+1} \in [0, \mu_{max}]. \tag{7.10}$$

(Em geral, $\bar{\lambda}^{k+1}$ e $\bar{\mu}^{k+1}$ serão as projeções de λ^{k+1} e μ^{k+1} em $[\lambda_{min}, \lambda_{max}]^m$ e $[0, \mu_{max}]^p$, respectivamente.)

Passo 4. *Atualizar parâmetro de penalidade*

Se

$$\max\{\|h(x^k)\|_\infty, \|V^k\|_\infty\} \leq \tau \max\{\|h(x^{k-1})\|_\infty, \|V^{k-1}\|_\infty\},$$

definir

$$\rho_{k+1} = \rho_k.$$

Se não, definir

$$\rho_{k+1} = \gamma \rho_k.$$

Passo 5. *Começar nova iteração*

Atualizar $k \leftarrow k + 1$. Voltar a Passo 2.

Observe que, tal como foi definido, o algoritmo não tem critério de parada. Isto é, mesmo que x^k seja uma solução global do problema original, o algoritmo "continua", neste caso repetindo o ponto x^k, para sempre. Naturalmente, isto é um artifício teórico pois, na prática, se estabelecem critérios para interromper a execução em um número finito, e moderado, de iterações.

7.1 Convergência a pontos admissíveis

Nesta seção vamos supor que o algoritmo não pára no Passo 2. Em outras palavras, sempre pode ser encontrado x^k satisfazendo (7.3)-(7.6).

Ao mesmo tempo, vamos supor que existe pelo menos um ponto limite da seqüência gerada pelo Algoritmo 7.1. Uma condição suficiente para isto é que $\Omega \neq \emptyset$ e existe $\varepsilon > 0$ tal que o conjunto

$$\{x \in I\!R^n \mid \underline{g}(x) \leq \varepsilon, \|\underline{h}(x)\| \leq \varepsilon\}$$

é limitado. Esta condição pode ser forçada acrescentando restrições artificiais no conjunto Ω.

Vamos investigar o status dos pontos limite de seqüências geradas pelo Algoritmo 7.1. Primeiro provaremos um resultado sobre a admissibilidade desses pontos limite. O Teorema 7.1 mostra que todo ponto limite é admissível, ou é um ponto KKT da soma de quadrados das inviabilidades, ou não satisfaz a condição CPLD com respeito às restrições Ω. Esta última alternativa é, na prática, pouco frequente porque o conjunto Ω costuma ser simples, de maneira que todos seus pontos geralmente satisfazem a condição CPLD.

Teorema 7.1. *Seja $\{x^k\}$ uma seqüência infinita gerada pelo Algoritmo 7.1. Seja x^* um ponto limite de $\{x^k\}$. Então, se a seqüência de parâmetros de penalidade $\{\rho_k\}$ é limitada, x^* é admissível. Em caso contrário, pelo menos uma das seguintes possibilidades acontece:*

(i) O ponto x^* é um ponto KKT do problema

$$\textit{Minimizar} \left[\sum_{i=1}^{m} h_i(x)^2 + \sum_{i=1}^{p} \max\{0, g_i(x)\}^2\right] \textit{ sujeita a } x \in \Omega.$$

(ii) x^* não satisfaz a condição CPLD associada a Ω.

Prova. Seja K uma subseqüência de $I\!N$ tal que

$$\lim_{k \in K} x^k = x^*.$$

Por (7.2), (7.4) e (7.6), temos que $\underline{g}(x^*) \le 0$ e $\underline{h}(x^*) = 0$. Portanto,

$$x^* \in \Omega.$$

Agora consideremos duas possibilidades:

(a) A seqüência $\{\rho_k\}$ é limitada.

(b) A seqüência $\{\rho_k\}$ não é limitada.

Analisemos primeiro o Caso (a). Neste caso, a partir de certa iteração, o parâmetro de penalidade não é mais atualizado. Portanto,

$$\lim_{k\to\infty} \|h(x^k)\| = \lim_{k\to\infty} \|V^k\| = 0.$$

Portanto,

$$h(x^*) = 0.$$

Mais ainda, se $g_j(x^*) > 0$ temos que $g_j(x^k) > c > 0$ para $k \in K$ suficientemente grande. Isto contradiz o fato de que $V_j^k \to 0$.

Logo,

$$g_i(x^*) \leq 0 \quad \text{para todo } i = 1, \ldots, p.$$

Logo, como $x^* \in \Omega$ e $h(x^*) = 0$, x^* é admissível.

Assim, a tese está provada no caso em que $\{\rho_k\}$ é limitada.

Consideremos agora o Caso (b). Ou seja, $\{\rho_k\}_{k \in K}$ não é limitada. Pela definição do Lagrangiano Aumentado 7.1 e a condição de Lagrange aproximada (7.3), temos:

$$\nabla f(x^k) + \sum_{i=1}^{m} (\bar{\lambda}_i^k + \rho_k h_i(x^k)) \nabla h_i(x^k) + \sum_{i=1}^{p} \max\{0, \bar{\mu}_i^k + \rho_k g_i(x^k)\} \nabla g_i(x^k)$$

$$+ \sum_{i=1}^{\underline{m}} v_i^k \nabla \underline{h}_i(x^k) + \sum_{i=1}^{\underline{p}} u_i^k \nabla \underline{g}_i(x^k) = \delta^k, \tag{7.11}$$

onde, como $\varepsilon_k \to 0$ (7.2),

$$\lim_{k \in K} \|\delta^k\| = 0.$$

Se $\underline{g}_j(x^*) < 0$, existe $k_1 \in I\!N$ tal que $\underline{g}_j(x^k) < -\varepsilon_k$ para todo $k \geq k_1, k \in K$. Portanto, pela complementaridade aproximada (7.5), $u_j^k = 0$ para todo $k \in K, k \geq k_1$.

Logo, por $x^* \in \Omega$ e (7.11), para todo $k \in K, k \geq k_1$ temos que

$$\nabla f(x^k) + \sum_{i=1}^{m} (\bar{\lambda}_i^k + \rho_k h_i(x^k)) \nabla h_i(x^k) + \sum_{i=1}^{p} \max\{0, \bar{\mu}_i^k + \rho_k g_i(x^k)\} \nabla g_i(x^k)$$

$$+ \sum_{i=1}^{\underline{m}} v_i^k \nabla \underline{h}_i(x^k) + \sum_{\underline{g}_i(x^*)=0} u_i^k \nabla \underline{g}_i(x^k) = \delta^k.$$

Dividindo por ρ_k obtemos:

$$\frac{\nabla f(x^k)}{\rho_k} + \sum_{i=1}^{m} \left(\frac{\bar{\lambda}_i^k}{\rho_k} + h_i(x^k) \right) \nabla h_i(x^k) + \sum_{i=1}^{p} \max\left\{0, \frac{\bar{\mu}_i^k}{\rho_k} + g_i(x^k)\right\} \nabla g_i(x^k)$$

$$+ \sum_{i=1}^{\underline{m}} \frac{v_i^k}{\rho_k} \nabla \underline{h}_i(x^k) + \sum_{\underline{g}_j(x^*)=0} \frac{u_j^k}{\rho_k} \nabla \underline{g}_j(x^k) = \frac{\delta^k}{\rho_k}.$$

Pelo Lema de Carathéodory, existem

$$\widehat{I}_k \subset \{1, \ldots, \underline{m}\}, \widehat{J}_k \subset \{j \mid \underline{g}_j(x^*) = 0\},$$

$$\widehat{v}_i^k, \;\; i \in \widehat{I}_k \;\; , \;\; \widehat{u}_j^k \geq 0, \;\; j \in \widehat{J}_k$$

tais que os vetores

$$\{\nabla \underline{h}_i(x^k)\}_{i\in \widehat{I}_k} \cup \{\nabla \underline{g}_j(x^k)\}_{j\in \widehat{J}_k}$$

são linearmente independentes e

$$\frac{\nabla f(x^k)}{\rho_k} + \sum_{i=1}^{m} \left(\frac{\bar{\lambda}_i^k}{\rho_k} + h_i(x^k)\right)\nabla h_i(x^k) + \sum_{i=1}^{p} \max\left\{0, \frac{\bar{\mu}_i^k}{\rho_k} + g_i(x^k)\right\}\nabla g_i(x^k)$$

$$+ \sum_{i\in \widehat{I}_k} \widehat{v}_i^k \nabla \underline{h}_i(x^k) + \sum_{i\in \widehat{J}_k} \widehat{u}_i^k \nabla \underline{g}_i(x^k) = \frac{\delta^k}{\rho_k}.$$

Como há um número finito de possíveis conjuntos $\widehat{I}_k, \widehat{J}_k$, existe um conjunto infinito de índices K_1 tal que

$$K_1 \underset{\infty}{\subset} \{k \in K \mid k \geq k_1\},$$

$$\widehat{I}_k = \widehat{I},$$

e

$$\widehat{J} = \widehat{J}_k \subset \{i \mid \underline{g}_i(x^*) = 0\} \tag{7.12}$$

para todo $k \in K_1$. Assim, para todo $k \in K_1$ obtemos:

$$\frac{\nabla f(x^k)}{\rho_k} + \sum_{i=1}^{m} \left(\frac{\bar{\lambda}_i^k}{\rho_k} + h_i(x^k)\right)\nabla h_i(x^k) + \sum_{i=1}^{p} \max\left\{0, \frac{\bar{\mu}_i^k}{\rho_k} + g_i(x^k)\right\}\nabla g_i(x^k)$$

$$+ \sum_{i\in \widehat{I}} \widehat{v}_i^k \nabla \underline{h}_i(x^k) + \sum_{i\in \widehat{J}} \widehat{u}_i^k \nabla \underline{g}_i(x^k) = \frac{\delta^k}{\rho_k}, \tag{7.13}$$

e os gradientes

$$\{\nabla \underline{h}_i(x^k)\}_{i\in \widehat{I}} \cup \{\nabla \underline{g}_i(x^k)\}_{i\in \widehat{J}} \tag{7.14}$$

são linearmente independentes para todo $k \in K_1$.

Consideramos, de novo, dois casos:

1. A seqüência $\{\|(\widehat{v}^k, \widehat{u}^k)\|, k \in K_1\}$ é limitada.

2. A seqüência $\{\|(\widehat{v}^k, \widehat{u}^k)\|, k \in K_1\}$ não é limitada.

Se a seqüência $\{\|(\widehat{v}^k, \widehat{u}^k)\|\}_{k\in K_1}$ é limitada, existe $(\widehat{v}, \widehat{u}), \widehat{u} \geq 0$ e um conjunto infinito de indices $K_2 \underset{\infty}{\subset} K_1$ tal que

$$\lim_{k\in K_2} (\widehat{v}^k, \widehat{u}^k) = (\widehat{v}, \widehat{u}).$$

Como $\{\rho_k\}$ é não-limitada, temos que $\rho_k \to \infty$. Logo, pela limitação de $\bar{\lambda}^k$ e $\bar{\mu}^k$, $\lim \bar{\lambda}_i^k/\rho_k = 0 = \lim \bar{\mu}_j^k/\rho_k$ para todo i, j. Como $\delta^k \to 0$, por (7.8), (7.10), tomando limites para $k \in K_2$ in (7.13), obtemos:

$$\sum_{i=1}^{m} h_i(x^*)\nabla h_i(x^*) + \sum_{i=1}^{p} \max\{0, g_i(x^*)\}\nabla g_i(x^*) + \sum_{i\in\widehat{I}} \widehat{v}_i \nabla \underline{h}_i(x^*) + \sum_{i\in\widehat{J}} \widehat{u}_i \nabla \underline{g}_i(x^*) = 0,$$

onde $\hat{J} \subset \{i \in \{1, \ldots, p\} \mid \underline{g}_i(x^*) = 0\}$. Portanto, por $x^* \in \Omega$ e (7.12), x^* é ponto KKT da soma de quadrados de inviabilidades, como enunciado na tese do teorema. (De fato, temos omitido o caso em que $\widehat{I} = \emptyset$ ou $\widehat{J} = \emptyset$, que deixamos a cargo do leitor.)

Finalmente, suponhamos que $\{\|(\widehat{v}^k, \widehat{u}^k)\|\}_{k\in K_1}$ é não-limitada. Seja $K_3 \underset{\infty}{\subset} K_1$ tal que $\lim_{k\in K_3} \|(\widehat{v}^k, \widehat{u}^k)\| = \infty$ e $(\widehat{v}, \widehat{u}) \neq 0, \widehat{u} \geq 0$ tal que

$$\lim_{k\in K_3} \frac{(\widehat{v}^k, \widehat{u}^k)}{\|(\widehat{v}^k, \widehat{u}^k)\|} = (\widehat{v}, \widehat{u}).$$

Dividindo ambos lados de (7.13) por $\|(\widehat{v}^k, \widehat{u}^k)\|$ e tomando limites para $k \in K_3$, obtemos:

$$\sum_{i\in\widehat{I}} \widehat{v}_i \nabla \underline{h}_i(x^*) + \sum_{j\in\widehat{J}} \widehat{u}_j \nabla \underline{g}_j(x^*) = 0.$$

Mas $\underline{g}_j(x^*) = 0$ para todo $j \in \widehat{J}$. Logo, por (7.14), x^* não satisfaz a condição CPLD associada com o conjunto Ω. Com isto, a prova está completa. **QED**

7.2 Convergência a pontos KKT

Em continuação provamos um resultado de otimalidade. Vimos que pontos limite de seqüências geradas pelo Algoritmo 7.1 podem ser admissíveis ou não. Essencialmente, o Teorema 7.1 diz que, se x^* não é admissível então é, muito provavelmente, minimizador local da inadmissibilidade. Agora estamos interessados no status de pontos limite admissíveis.

Provaremos que, sob CPLD, os pontos limite são KKT do problema original. Antes, provaremos um resultado interessante que independe da CPLD. Este resultado nos diz que, se uma restrição de desigualdade se verifica estritamente no ponto limite, então a estimativa do multiplicador associado se anula, para k suficientemente grande.

Proposição 7.1 *Seja $\{x^k\}_{k\in\mathbb{N}}$ uma seqüência gerada pelo Algoritmo 7.1. Suponhamos que x^* é um ponto limite (não necessariamente admissível!) desta seqüência* *e* $K \underset{\infty}{\subset} I\!N$ *é tal que* $\lim_{k \in K} x^k = x$. *Então, para* $k \in K$ *suficientemente grande,*

$$g_i(x^*) < 0 \Rightarrow \mu_i^{k+1} = 0 \text{ e } \underline{g}_i(x^*) < 0 \Rightarrow u_i^k = 0. \tag{7.15}$$

Prova. Pelos critérios de parada do subproblema, que definem x^k no Algoritmo 7.1, temos que para todo $k \in K$, existem $u^k \in I\!R_+^{\underline{p}}$, $\delta^k \in I\!R^n$ tal que $\|\delta^k\| \leq \varepsilon_k$ e

$$\nabla f(x^k) + \sum_{i=1}^{m} \lambda_i^{k+1} \nabla h_i(x^k) + \sum_{i=1}^{p} \mu_i^{k+1} \nabla g_i(x^k) + \sum_{i=1}^{\underline{m}} v_i^k \nabla \underline{h}_i(x^k) + \sum_{i=1}^{\underline{p}} u_i^k \nabla \underline{g}_i(x^k) = \delta^k. \tag{7.16}$$

Por (7.9), $\mu^{k+1} \in I\!R_+^p$ para todo $k \in I\!N$.

Suponhamos que $\underline{g}_i(x^*) < 0$. Então, existe $k_1 \in I\!N$ tal que $\forall k \in K, k \geq k_1, \underline{g}_i(x^k) < -\varepsilon_k$. Então, por (7.5),

$$u_i^k = 0 \quad \forall k \in K, k \geq k_1.$$

Provemos agora que uma propriedade similar se verifica quando $g_i(x^*) < 0$. Neste caso, existe $k_2 \geq k_1$ tal que

$$g_i(x^k) < c < 0 \ \ \forall k \in K, k \geq k_2.$$

Consideramos dois casos:

1. $\{\rho_k\}$ não é limitada.
2. $\{\rho_k\}$ é limitada.

No primeiro caso, temos que $\lim_{k \in K} \rho_k = \infty$. Como $\{\bar{\mu}_i^k\}$ é limitada, existe $k_3 \geq k_2$ tal que, para todo $k \in K, k \geq k_3$,

$$\bar{\mu}_i^k + \rho_k g_i(x^k) < 0.$$

Pela definição de μ^{k+1} isto implica que

$$\mu_i^{k+1} = 0 \quad \forall k \in K, k \geq k_3.$$

Consideremos agora o caso em que $\{\rho_k\}$ é limitada. Neste caso,

$$\lim_{k \to \infty} V_i^k = 0.$$

Portanto, como $g_i(x^k) < c < 0$ para $k \in K$ suficientemente grande,

$$\lim_{k \in K} \bar{\mu}_i^k = 0.$$

Logo, para $k \in K$ suficientemente grande,

$$\bar{\mu}_i^k + \rho_k g_i(x^k) < 0.$$

Pela definição de μ^{k+1}, existe $k_4 \in I\!N$ tal que $\mu_i^{k+1} = 0$ para $k \in K, k \geq k_4$.

Portanto, existe $k_5 \geq \max\{k_3, k_4\}$ tal que para todo $k \in K$, $k \geq k_5$,

$$g_i(x^*) < 0 \Rightarrow \mu_i^{k+1} = 0 \text{ e } \underline{g}_i(x^*) < 0 \Rightarrow u_i^k = 0,$$

como queriamos provar. **QED**

Teorema 7.2. *Seja $\{x^k\}_{k \in I\!N}$ uma seqüência gerada pelo Algoritmo 7.1. Suponhamos que x^* é um ponto limite admissível que satisfaz CPLD com respeito a todas as restrições do problema original. Então , x^* é ponto KKT do problema original.*

Prova. Pela Proposição 7.1, existem $K \underset{\infty}{\subset} I\!N$, $k_5 \in I\!N$ tais que para todo $k \in K, k \geq k_5$, temos:

$$\nabla f(x^k) + \sum_{i=1}^{m} \lambda_i^{k+1} \nabla h_i(x^k) + \sum_{g_i(x^*)=0} \mu_i^{k+1} \nabla g_i(x^k)$$

$$+ \sum_{i=1}^{\underline{m}} v_i^k \nabla \underline{h}_i(x^k) + \sum_{\underline{g}_j(x^*)=0} u_j^k \nabla \underline{g}_j(x^k) = \delta^k, \tag{7.17}$$

com $\mu^{k+1} \in I\!R_+^p, u^k \in I\!R_+^{\underline{p}}$ e $\|\delta^k\| \leq \varepsilon_k$.

Pelo Lema de Carathéodory, para todo $k \in K, k \geq k_5$, existem

$$\widehat{I}_k \subset \{1, \ldots, m\}, \widehat{J}_k \subset \{j \mid g_j(x^*) = 0\}, \check{I}_k \subset \{1, \ldots, \underline{m}\}, \check{J}_k \subset \{j \mid \underline{g}_j(x^*) = 0\},$$

$$\widehat{\lambda}_i^k \; \forall i \in \widehat{I}_k, \widehat{\mu}_j^k \geq 0 \; \forall j \in \widehat{J}_k, \widehat{v}_i^k \; \forall i \in \check{I}_k, \widehat{u}_j^k \geq 0 \; \forall j \in \check{J}_k$$

tais que os vetores

$$\{\nabla h_i(x^k)\}_{i \in \widehat{I}_k} \cup \{\nabla g_i(x^k)\}_{i \in \widehat{J}_k} \cup \{\nabla \underline{h}_i(x^k)\}_{i \in \check{I}_k} \cup \{\nabla \underline{g}_i(x^k)\}_{i \in \check{J}_k}$$

são linearmente independentes e

$$\nabla f(x^k) + \sum_{i \in \widehat{I}_k} \widehat{\lambda}_i^k \nabla h_i(x^k) + \sum_{i \in \widehat{J}_k} \widehat{\mu}_i^k \nabla g_i(x^k)$$

$$+ \sum_{i \in \check{I}_k} \widehat{v}_i^k \nabla \underline{h}_i(x^k) + \sum_{j \in \check{J}_k} \widehat{u}_j^k \nabla \underline{g}_j(x^k) = \delta^k.$$

Como o número de possíveis conjuntos de índices $\widehat{I}_k, \widehat{J}_k, \check{I}_k, \check{J}_k$ é finito, existe $K_1 \underset{\infty}{\subset} \{k \in K \mid k \geq k_5\}$ tal que

$$\widehat{I}_k = \widehat{I}, \widehat{J}_k = \widehat{J}, \check{I}_k = \check{I}, \check{J}_k = \check{J},$$

para todo $k \in K_1$.

Logo,

$$\begin{aligned} &\nabla f(x^k) + \textstyle\sum_{i\in\widehat{I}} \widehat{\lambda}_i^k \nabla h_i(x^k) + \sum_{i\in\widehat{J}} \widehat{\mu}_i^k \nabla g_i(x^k) \\ &+ \textstyle\sum_{i\in\check{I}} \widehat{v}_i^k \nabla \underline{h}_i(x^k) + \sum_{j\in\check{J}} \widehat{u}_j^k \nabla \underline{g}_j(x^k) = \delta^k \end{aligned} \tag{7.18}$$

e os vetores

$$\{\nabla h_i(x^k)\}_{i\in\widehat{I}} \cup \{\nabla g_i(x^k)\}_{i\in\widehat{J}} \cup \{\nabla \underline{h}_i(x^k)\}_{i\in\check{I}} \cup \{\nabla \underline{g}_i(x^k)\}_{i\in\check{J}} \tag{7.19}$$

são linearmente independentes para todo $k \in K_1$.

Definimos

$$S_k = \max\{\max\{|\widehat{\lambda}_i^k|, i \in \widehat{I}\}, \max\{\widehat{\mu}_i^k, i \in \widehat{J}\}, \max\{|\widehat{v}_i^k|, i \in \check{I}\}, \max\{\widehat{u}_i^k, i \in \check{J}\}\}.$$

Consideremos duas possibilidades:

(a) $\{S_k\}_{k\in K_1}$ tem uma subseqüência limitada.

(b) $\lim_{k\in K_1} S_k = \infty$.

Se $\{S_k\}_{k\in K_1}$ admite uma subseqüência limitada, existe $K_2 \underset{\infty}{\subset} K_1$ tal que

$$\lim_{k\in K_2} \widehat{\lambda}_i^k = \widehat{\lambda}_i,$$

$$\lim_{k\in K_2} \widehat{\mu}_i^k = \widehat{\mu}_i \geq 0,$$

$$\lim_{k\in K_2} \widehat{v}_i^k = \widehat{v}_i,$$

$$\lim_{k\in K_2} \widehat{u}_i^k = \widehat{u}_i \geq 0.$$

Por (7.2) e o fato de que $\|\delta^k\| \leq \varepsilon_k$, tomando limites em (7.18) para $k \in K_2$, obtemos:

$$\nabla f(x^*) + \sum_{i\in\widehat{I}} \widehat{\lambda}_i \nabla h_i(x^*) + \sum_{i\in\widehat{J}} \widehat{\mu}_i \nabla g_i(x^*) + \sum_{i\in\check{I}} \widehat{v}_i \nabla \underline{h}_i(x^*) + \sum_{j\in\check{J}} \widehat{u}_j \nabla \underline{g}_j(x^*) = 0,$$

com $\widehat{\mu}_i \geq 0, \widehat{u}_i \geq 0$. Como x^* é admissível, temos que x^* é um ponto KKT do problema original.

Suponhamos agora que $\lim_{k\in K_1} S_k = \infty$. Dividindo ambos lados de (7.18) por S_k obtemos:

$$\begin{aligned} &\tfrac{\nabla f(x^k)}{S_k} + \textstyle\sum_{i\in\widehat{I}} \frac{\widehat{\lambda}_i^k}{S_k} \nabla h_i(x^k) + \sum_{i\in\widehat{J}} \frac{\widehat{\mu}_i^k}{S_k} \nabla g_i(x^k) \\ &+ \textstyle\sum_{i\in\check{I}} \frac{\widehat{v}_i^k}{S_k} \nabla \underline{h}_i(x^k) + \sum_{j\in\check{J}} \frac{\widehat{u}_j^k}{S_k} \nabla \underline{g}_j(x^k) = \frac{\delta^k}{S_k}, \end{aligned} \tag{7.20}$$

onde
$$\left|\frac{\widehat{\lambda}_i^k}{S_k}\right| \leq 1, \left|\frac{\widehat{\mu}_i^k}{S_k}\right| \leq 1, \left|\frac{\widehat{v}_i^k}{S_k}\right| \leq 1, \left|\frac{\widehat{u}_j^k}{S_k}\right| \leq 1.$$
Portanto, existe $K_3 \underset{\infty}{\subset} K_1$ tal que

$$\lim_{k\in K_3} \frac{\widehat{\lambda}_i^k}{S_k} = \widehat{\lambda}_i, \lim_{k\in K_3} \frac{\widehat{\mu}_i^k}{S_k} = \widehat{\mu}_i \geq 0, \lim_{k\in K_3} \frac{\widehat{v}_i^k}{S_k} = \widehat{v}_i, \lim_{k\in K_3} \frac{\widehat{u}_j^k}{S_k} = \widehat{u}_j \geq 0.$$

Tomando limites em ambos lados de (7.20) para $k \in K_3$, obtemos:

$$\sum_{i\in\widehat{I}} \widehat{\lambda}_i \nabla h_i(x^*) + \sum_{i\in\widehat{J}} \widehat{\mu}_i \nabla g_i(x^*) + \sum_{i\in\check{I}} \widehat{v}_i \nabla \underline{h}_i(x^*) + \sum_{j\in\check{J}} \widehat{u}_j \nabla \underline{g}_j(x^*) = 0.$$

Mas o módulo de pelo menos um dos coeficientes $\widehat{\lambda}_i, \widehat{\mu}_i, \widehat{v}_i, \widehat{u}_i$ é igual a 1. Logo, pela condição CPLD, os gradientes

$$\{\nabla h_i(x)\}_{i\in\widehat{I}} \cup \{\nabla g_i(x)\}_{i\in\widehat{J}} \cup \{\nabla \underline{h}_i(x)\}_{i\in\check{I}} \cup \{\nabla \underline{g}_i(x)\}_{i\in\check{J}}$$

são linearmente dependentes em uma vizinhança de x^*. Isto contradiz (7.19). Portanto, o teorema está provado. **QED**

EXERCÍCIOS

1. Suponha as hipóteses do Teorema 7.2. Seja x^* um ponto admissível tal que $\lim_{k\in K} x^k = x^*$ e suponha que x^* satisfaz MF. Prove que o conjunto $\{\lambda^{k+1}, \mu^{k+1}, v^k, u^k\}$ é limitado para $k \in K$.

2. Com base nos resultados teóricos deste capítulo, sugira critérios práticos de parada para o método de Lagrangiano Aumentado.

3. Suponha que o conjunto Ω é uma caixa. Defina os subproblemas do método de Lagrangiano Aumentado usando o gradiente projetado contínuo da função Lagrangiano Aumentado, com o critério interno de parada natural, baseado na norma do gradiente projetado contínuo. Mostre que fazer isto é equivalente a fazer o que fizemos neste capítulo, com a definição funcional de Ω.

4. Invente exemplos com um único minimizador global x^* e com as seguintes situações: (a) x^* não é CPLD; (b) x^* é CPLD mas não é MF; (c) x^* é MF mas não é regular. De acordo com a teoria deste capítulo, diga o que você acha que deverá acontecer quando aplicarmos `Algencan` para resolver os problemas. Rode `Algencan` e confira a exatidão de suas previsões. Explique as eventuais diferenças.

Capítulo 8

Limitação do Parâmetro de Penalidade

Quando o parâmetro de penalidade é grande, os subproblemas do Método de Lagrangiano Aumentado são difíceis de resolver. A razão para isto é que, ao ser multiplicado por um número muito grande, o termo penalizado domina a função objetivo do problema. Assim, $f(x)$ funciona como uma pequena perturbação do segundo termo e o algoritmo que resolve o subproblema desprezará tal perturbação. Em ponto flutuante, a soma de um número muito grande com um número relativamente pequeno é o número grande. Portanto, quando se minimiza uma função de tipo Lagrangiano Aumentado com um parâmetro grande de penalidade, a tendência é que, na maioria das iterações, o método interno se comporte como se o termo $f(x)$ não existisse. Os conjuntos de nível do Lagrangiano Aumentado, neste caso, costumam ser vales alongados que acompanham, aproximadamente, os conjuntos de nível do termo penalizado.

É vital, em conseqüência, que os métodos de Lagrangiano Aumentado tenham a propriedade de que o parâmetro de penalidade não cresça indefinidamente. Nos termos do Algoritmo 7.1, queremos que o teste do Passo 4 tenha resultado positivo para todo k suficientemente grande.

Neste capítulo nos propomos provar que, de fato, o parâmetro de penalidade fica limitado com certa freqüência, isto é, sob condições suficientes que, muitas vezes, são satisfeitas nos problemas práticos.

8.1 Restrições de Igualdade

Primeiro, nos limitaremos a um caso particular ($p = 0$, $\Omega = I\!R^n$) que, no entanto, abrange as idéias fundamentais da prova geral. Em todo caso, é bom que o leitor interessado no teorema mais geral esteja previamente familiarizado com este caso.

Assim, o problema PNL considerado nesta seção, será:

$$\text{Minimizar } f(x) \text{ sujeita a } h(x) = 0,$$

com $h : I\!R^n \to I\!R^m, f : I\!R^n \to I\!R$.

Conseqüentemente, o Algoritmo de Lagrangiano Aumentado ficará simplificado da seguinte maneira:

Algoritmo 8.1.

Seja $x^0 \in I\!R^n$ um ponto inicial arbitrário.

Os parâmetros para a execução do algoritmo são:

$$\tau \in [0, 1), \gamma > 1,$$

$$-\infty < \lambda_{min} < \lambda_{max} < \infty,$$

$$\rho_1 > 0,$$

$$\bar{\lambda}_i^1 \in [\lambda_{\min}, \lambda_{max}] \; \forall i = 1, \ldots, m,$$

Por último, $\{\varepsilon_k\} \subset I\!R_+$ será uma seqüência de parâmetros de tolerância tal que

$$\lim_{k \to \infty} \varepsilon_k = 0.$$

Passo 1. *Inicialização*

$k \leftarrow 1$.

Passo 2. *Resolução do subproblema*

Calcular (se for possível) $x^k \in I\!R^n$ tal que

$$\|\nabla L_{\rho_k}(x^k, \bar{\lambda}^k)\| \leq \varepsilon_k,$$

Se não é possível encontrar x^k cumprindo essa propriedade, parar a execução do algoritmo.

Passo 3. *Estimar multiplicadores*

Para todo $i = 1, \ldots, m$, calcular

$$\lambda_i^{k+1} = \bar{\lambda}_i^k + \rho_k h_i(x^k)$$

e

$$\bar{\lambda}_i^{k+1} = \max\{\lambda_{min}, \min\{\lambda_{max}, \lambda_i^{k+1}\}\}.$$

Passo 4. *Atualizar o parâmetro de penalidade*

Se

$$\|h(x^k)\|_\infty \leq \tau \|h(x^{k-1})\|_\infty,$$

definir

$$\rho_{k+1} = \rho_k.$$

Em caso contrário, definir

$$\rho_{k+1} = \gamma \rho_k.$$

Passo 5. *Começar uma nova iteração*
Atualizar $k \leftarrow k+1$. Voltar a Passo 2.

8.1.1 Hipóteses

O resultado fundamental depende de seis suposições:

1. A seqüência $\{x^k\}$ gerada pelo Algoritmo 8.1 converge a x^*.
2. O ponto x^* é admissível (ou seja, $h(x^*) = 0$).
3. O ponto x^* é regular (ou seja, a matriz $\nabla h(x^*)$ tem posto igual a m).
4. Os multiplicadores λ_i^* pertencem a $(\lambda_{min}, \lambda_{max})$ para todo $i = 1, \ldots, m$. (Observe que a existência de λ^* satisfazendo $\nabla f(x^*) + \nabla h(x^*)\lambda^*$ está garantida pela regularidade de x^*.)
5. As funções f e h admitem derivadas segundas contínuas em uma vizinhança de x^*. (Os operadores ∇ e ∇^2 denotarão sempre derivadas em relação a x.)
6. Seja $Z \in I\!R^{n \times (n-m)}$ uma matriz cujas colunas formam uma base do núcleo de $\nabla h(x^*)^T$. Então, $Z^T \nabla^2 L_0(x^*, \lambda^*) Z$ é definida positiva.

Não temos condições suficientes sobre o problema que garantam que toda a seqüência é convergente. O que provamos em capítulos anteriores é que o algoritmo de Lagrangiano Aumentado é convergente no sentido de subseqüências. Em princípio, um algoritmo de Lagrangiano Aumentado pode pular entre soluções muito afastadas de sucessivos subproblemas e a seqüência gerada pode ter mais de um ponto de acumulação. Entretanto, a hipótese de convergência colocada aqui não é muito arbitrária por uma razão algorítmica: em geral, o ponto inicial para o subalgoritmo que resolve um subproblema é escolhido como a solução do subproblema anterior. Desta maneira, soluções de subproblemas consecutivos costumam ficar perto umas de outras e a oscilação entre diferentes pontos de acumulação é improvável.

Sob a hipótese de que o limite x^* é regular, a existência dos multiplicadores de Lagrange λ^* está garantida pois, como vimos, a regularidade é uma condição de qualificação. A sexta hipótese colocada acima se denomina Condição Suficiente de Segunda Ordem na literatura de Otimização [5, 13]. Como seu nome indica, sua satisfação em um ponto KKT regular implica que o ponto é um minimizador local.

8.1.2 Teorema principal

As hipóteses introduzidas na Seção anterior serão usadas nesta seção sem prévio aviso.

Lema 8.1. *A seqüência* $\{\lambda^k\}$ *converge a* λ^*.

Prova. Suponhamos que λ^k não converge a λ^*. Portanto, λ^{k+1} não converge a λ^*. Logo, existe $c > 0$ e um conjunto infinito de índices $K \subset I\!N$ tal que

$$\|\lambda^{k+1} - \lambda^*\| \geq c \; \forall k \in K. \tag{8.1}$$

Pelo critério de resolução aproximada dos subproblemas e a fórmula de λ^{k+1} temos:

$$\lim_{k\to\infty} \nabla f(x^k) + \nabla h(x^k)\lambda^{k+1} = 0.$$

Suponhamos primeiro que a subseqüência $\{\lambda^{k+1}\}_{k\in K}$ esteja limitada. Neste caso, ela tem uma subseqüência convergente a, digamos, $\widehat{\lambda}$. Portanto, passando ao limite para essa subseqüência,

$$\nabla f(x^*) + \nabla h(x^*)\widehat{\lambda} = 0.$$

Como a matriz $\nabla h(x^*)$ tem posto completo, devemos ter $\widehat{\lambda} = \lambda^*$. Isto contradiz (8.1).

Suponhamos que a subseqüência $\{\lambda^{k+1}\}_{k\in K}$ não está limitada. Então, dividindo por $\|\lambda^{k+1}\|$, e tomando outra subseqüencia adequada, verificamos que o ponto x^* não é regular, o que contradiz a hipótese. **QED**

Lema 8.2. *Para* k *suficientemente grande,* $\bar{\lambda}^k = \lambda^k$.

Prova. Pelo Lema 8.1, λ^k tende a λ^*. Mas, como λ^* pertence a $(\lambda_{min}, \lambda_{max})^m$, temos que $\lambda^{k+1} \in [\lambda_{min}, \lambda_{max}]^m$ para k suficientemente grande. Portanto, para k suficientemente grande, $\bar{\lambda}^{k+1} = \lambda^{k+1}$. **QED**

Lema 8.3. *Existem* $c > 0$, $\bar{\rho} > 0$, $\delta > 0$ *tais que, se* $\pi \in [0, 1/\bar{\rho}]$, $\|x - x^*\| \leq \delta$ *e* $\|\lambda - \lambda^*\| \leq \delta$, *a matriz*

$$\int_0^1 \begin{pmatrix} \nabla^2 L_0(x^* + t(x - x^*), \lambda^* + t(\lambda - \lambda^*)) & \nabla h(x^* + t(x - x^*)) \\ \nabla h(x^* + t(x - x^*))^T & -\pi I \end{pmatrix} dt$$

é não-singular e

$$\left\| \left[\int_0^1 \begin{pmatrix} \nabla^2 L_0(x^* + t(x - x^*), \lambda^* + t(\lambda - \lambda^*)) & \nabla h(x^* + t(x - x^*)) \\ \nabla h(x^* + t(x - x^*))^T & -\pi I \end{pmatrix} dt \right]^{-1} \right\| \leq c.$$

Prova. Vejamos primeiro que a matriz é não-singular se $\pi = 0$, $x = x^*$, $\lambda = \lambda^*$. Suponhamos que $u^1 \in I\!R^n, u^2 \in I\!R^m$ são tais que

$$\begin{pmatrix} \nabla^2 L_0(x^*, \lambda^*) & \nabla h(x^*) \\ \nabla h(x^*)^T & 0 \end{pmatrix} \begin{pmatrix} u^1 \\ u^2 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}.$$

Logo, u^1 está no núcleo de $\nabla h(x^*)^T$, portanto, existe $v \in I\!R^{n-m}$ tal que

$$u^1 = Zv.$$

Portanto,

$$\nabla^2 L_0(x^*, \lambda^*) Zv + \nabla h(x^*) u^2 = 0.$$

Pre-multiplicando por $v^T Z^T$ obtemos:

$$v^T Z^T \nabla^2 L_0(x^*, \lambda^*) Zv + v^T Z^T \nabla h(x^*) u^2 = 0.$$

Pela definição de Z, temos que $Z^T \nabla h(x^*) = 0$, portanto:

$$v^T Z^T \nabla^2 L_0(x^*, \lambda^*) Zv = 0.$$

Mas, pelas Hipóteses, $Z^T \nabla^2 L_0(x^*, \lambda^*) Z$ é definida positiva, logo $v = 0$. Isto implica que $u^1 = 0$. Em conseqüência,

$$\nabla h(x^*) u^2 = 0.$$

Como as colunas de $\nabla h(x^*)$ são, por hipótese, linearmente independentes, deduzimos que $u^2 = 0$.

Portanto, a matriz $\begin{pmatrix} \nabla^2 L_0(x^*, \lambda^*) & \nabla h(x^*) \\ \nabla h(x^*)^T & -\pi I \end{pmatrix}$ é não-singular se $\pi = 0$.

Mas

$$\begin{pmatrix} \nabla^2 L_0(x^*, \lambda^*) & \nabla h(x^*) \\ \nabla h(x^*)^T & -\pi I \end{pmatrix}$$

$$= \int_0^1 \begin{pmatrix} \nabla^2 L_0(x^* + t(x^* - x^*), \lambda^* + t(\lambda^* - \lambda^*)) & \nabla h(x^* + t(x^* - x^*)) \\ \nabla h(x^* + t(x^* - x^*))^T & -\pi I \end{pmatrix} dt.$$

Logo, pela continuidade das derivadas segundas e da inversa de matrizes, obtemos a tese do Lema. **QED**

Lema 8.4. *Seja $\bar\rho$ como no Lema 8.3. Suponhamos que existe $k_0 \in I\!N$ tal que $\rho_k \geq \bar\rho$ para todo $k \geq k_0$. Então, existe $M > 0$ tal que, para todo $k \geq k_0$,*

$$\|x^k - x^*\| \leq M \max\left\{ \frac{\|\bar\lambda^k - \lambda^*\|}{\rho_k}, \|\nabla L_{\rho_k}(x^k, \bar\lambda^k)\| \right\}$$

e

$$\|\lambda^{k+1}-\lambda^*\| \leq M \max\left\{\frac{\|\bar{\lambda}^k-\lambda^*\|}{\rho_k}, \|\nabla L_{\rho_k}(x^k,\bar{\lambda}^k)\|\right\}.$$

Prova. Definimos, para todo $k \in I\!N$,

$$z^k = \nabla L_{\rho_k}(x^k,\bar{\lambda}^k) \in I\!R^n,$$

$$y^k = \frac{\lambda^*-\bar{\lambda}^k}{\rho_k} \in I\!R^m,$$

$$\pi_k = \frac{1}{\rho_k} \in I\!R_+.$$

Observe que, pela definição de λ^{k+1},

$$z^k = \nabla L_0(x^k,\lambda^{k+1}).$$

Logo, para todo $k \in I\!N$,

$$z^k = \nabla f(x^k) + \nabla h(x^k)\lambda^{k+1},$$

$$y^k = h(x^k) + (\lambda^*-\lambda^{k+1})\pi_k.$$

Definimos, para todo $\pi \in [0, 1/\bar{\rho}]$,

$$F_\pi : I\!R^{n+m} \to I\!R^{n+m}$$

por

$$F_\pi(x,\lambda) = \begin{pmatrix} \nabla f(x) + \nabla h(x)\lambda \\ h(x) + (\lambda^*-\lambda)\pi \end{pmatrix}.$$

Claramente,

$$F_{\pi_k}(x^k,\lambda^{k+1}) = \begin{pmatrix} z^k \\ y^k \end{pmatrix} \tag{8.2}$$

e, pelas hipóteses,

$$F_{\pi_k}(x^*,\lambda^*) = 0$$

para todo $k \in I\!N$.

Portanto,

$$F_{\pi_k}(x^k,\lambda^{k+1}) - F_{\pi_k}(x^*,\lambda^*) = \begin{pmatrix} z^k \\ y^k \end{pmatrix}.$$

Mas, pelo Teorema Fundamental do Cálculo Integral,

$$F_{\pi_k}(x^k,\lambda^{k+1}) - F_{\pi_k}(x^*,\lambda^*)$$

$$= \left[\int_0^1 \begin{pmatrix} \nabla^2 L_0(x^*+t(x^k-x^*), \lambda^*+t(\lambda^{k+1}-\lambda^*)) & \nabla h(x^*+t(x^k-x^*)) \\ \nabla h(x^*+t(x^k-x^*))^T & -\pi_k I \end{pmatrix} dt\right] \begin{pmatrix} x^k-x^* \\ \lambda^{k+1}-\lambda^* \end{pmatrix}.$$

Logo, pelo Lema 8.3, se $\rho_k \geq \bar{\rho}$, e k é suficientemente grande, teremos $\pi_k \leq 1/\bar{\rho}$, $\|x^k - x^*\| \leq \delta$ e $\|\lambda^{k+1} - \lambda^*\| \leq \delta$. Portanto,

$$\left\| \begin{pmatrix} x^k - x^* \\ \lambda^{k+1} - \lambda^* \end{pmatrix} \right\| \leq c \left\| \begin{pmatrix} z^k \\ y^k \end{pmatrix} \right\|.$$

Isto implica a tese do lema. **QED**

Teorema 8.1. *Suponhamos, além das hipóteses gerais desta seção, que existe uma seqüência* $\{\eta_k\}$ *que converge a zero, tal que*

$$\varepsilon_k \leq \eta_k \|h(x^k)\|_\infty$$

para todo $k \in I\!N$. *Então, a seqüência de parâmetros de penalidade* $\{\rho_k\}$ *está limitada.*

Prova. Suponhamos que $\lim_{k\to\infty} \rho_k = \infty$. Como $h(x^*) = 0$, pela continuidade das primeiras derivadas de h, existe $L > 0$ tal que para todo $k \in I\!N$,

$$\|h(x^k)\|_\infty \leq L\|x^k - x^*\|.$$

Pela hipótese, o Lema 8.4 e o fato de que $\bar{\lambda}^k = \lambda^k$ para k suficientemente grande, temos que

$$\|h(x^k)\|_\infty \leq LM \max\left\{ \frac{\|\lambda^k - \lambda^*\|_\infty}{\rho_k}, \eta_k \|h(x^k)\|_\infty \right\}.$$

Como $\eta_k \to 0$, isto implica que

$$\|h(x^k)\|_\infty \leq LM \frac{\|\lambda^k - \lambda^*\|_\infty}{\rho_k} \tag{8.3}$$

para k suficientemente grande.

Pela fórmula de λ^{k+1}, temos que

$$\lambda^k = \lambda^{k-1} + \rho_{k-1} h(x^{k-1}).$$

Portanto,

$$\|h(x^{k-1})\|_\infty = \frac{\|\lambda^k - \lambda^{k-1}\|_\infty}{\rho_{k-1}} \geq \frac{\|\lambda^{k-1} - \lambda^*\|_\infty}{\rho_{k-1}} - \frac{\|\lambda^k - \lambda^*\|_\infty}{\rho_{k-1}}. \tag{8.4}$$

Pelo Lema 8.4, a hipótese deste teorema e por $\bar{\lambda}^k = \lambda^k$, temos:

$$\|\lambda^k - \lambda^*\|_\infty \leq M\left(\frac{\|\lambda^{k-1} - \lambda^*\|_\infty}{\rho_{k-1}} + \eta_{k-1} \|h(x^{k-1})\|_\infty \right).$$

Isto implica que

$$\frac{\|\lambda^{k-1} - \lambda^*\|_\infty}{\rho_{k-1}} \geq \frac{\|\lambda^k - \lambda^*\|_\infty}{M} - \eta_{k-1}\|h(x^{k-1})\|_\infty.$$

Logo, por (8.4), para $\rho_{k-1} \geq 2M$,

$$(1 + \eta_{k-1})\|h(x^{k-1})\|_\infty \geq \|\lambda^k - \lambda^*\|_\infty \Big(\frac{1}{M} - \frac{1}{\rho_{k-1}}\Big) \geq \frac{1}{2M}\|\lambda^k - \lambda^*\|_\infty.$$

Portanto,

$$\|\lambda^k - \lambda^*\|_\infty \leq 3M\|h(x^{k-1})\|_\infty$$

para k suficientemente grande. Por (8.3), temos que

$$\|h(x^k)\|_\infty \leq \frac{3LM^2}{\rho_k}\|h(x^{k-1})\|_\infty.$$

Logo, como $\rho_k \to \infty$, existe $k_1 \in I\!N$ tal que

$$\|h(x^k)\|_\infty \leq \tau\|h(x^{k-1})\|_\infty$$

para todo $k \geq k_1$. Isto implica que $\rho_{k+1} = \rho_k$ para todo $k \geq k_1$. Em conseqüência, a seqüência $\{\rho_k\}$ está limitada. **QED**

8.2 Restrições de Desigualdade

Consideraremos o problema PNL na seguinte forma:

$$\text{Minimizar } f(x), \text{ sujeita a } h(x) = 0, g(x) \leq 0,$$

onde, como sempre, $f : I\!R^n \to I\!R, h : I\!R^n \to I\!R^m, g : I\!R^n \to I\!R^p$.

O Algoritmo de Lagrangiano aumentado será o Algoritmo 7.1, com $\Omega = I\!R^n$. Como no caso de restrições de igualdade, a limitação dos parâmetros de penalidade dependerá de um conjunto de hipóteses básicas.

8.2.1 Hipóteses

1. A seqüência $\{x^k\}$ gerada pelo Algoritmo 7.1 converge a x^*.

2. O ponto x^* é admissível (ou seja, $h(x^*) = 0$, e $g(x^*) \leq 0$).

3. O ponto x^* é regular. Ou seja, o conjunto formado pelos gradientes das restrições de igualdade e os gradientes das restrições de desigualdade tais que $g_i(x^*) = 0$ é linearmente independente. Como no caso de restrições de igualdade, isto implica que as condições KKT se verificam em x^*. Portanto, existem multiplicadores de Lagrange $\lambda^* \in I\!R^m$, $\mu^* \in I\!R^p_+$ associados a estas restrições.

4. O multiplicador λ_i^* pertence a $(\lambda_{min}, \lambda_{max})$ para todo $i = 1, \ldots, m$. Analogamente, cada multiplicador μ_i^* pertence a $[0, \mu_{max})$ para todo $i = 1, \ldots, p$.

5. As funções f, h e g admitem derivadas segundas contínuas em uma vizinhança de x^*. (Como no caso de restrições de igualdade, os operadores ∇ e ∇^2 denotarão sempre derivadas em relação a x.)

6. Para todo $i = 1, \ldots, p$ tal que $g_i(x^*) = 0$, temos $\mu_i^* > 0$. Esta condição se denomina *Complementaridade Estrita.*

7. Seja $A \in I\!R^{(m+q)\times n}$ a matriz cujas linhas são os gradientes das restrições de igualdade seguidas dos gradientes das q restrições ativas $(g_i(x^*) = 0)$ em x^*. Seja $Z \in I\!R^{n\times(n-m-q)}$ uma matriz cujas colunas formam uma base do núcleo de A. Então, $Z^T\nabla^2 L_0(x^*, \lambda^*, \mu^*)Z$ é definida positiva.

A complementaridade estrita junto com a última hipótese definida acima definem uma condição suficiente para minimizador local do problema PNL em consideração.

8.2.2 Teorema de limitação

Teorema 8.2. *Suponhamos, além das hipóteses gerais desta seção, que existe uma seqüência $\{\eta_k\}$ que converge a zero, tal que*

$$\varepsilon_k \leq \eta_k \max\{\|h(x^k)\|_\infty, \|V^k\|_\infty\}$$

para todo $k \in I\!N$. Então, a seqüência de parâmetros de penalidade $\{\rho_k\}$ está limitada.

Prova. A técnica para provar este teorema consiste em reduzir o problema e o próprio Algoritmo 7.1 ao caso com restrições de igualdade. Sem perda de generalidade, suponhamos que as restrições de desigualdade ativas em x^* são as q primeiras, de modo que

$$g_i(x^*) = 0 \text{ se } i \leq q, \;\; g_i(x^*) < 0 \text{ se } i > q.$$

Consideremos o problema auxiliar:

$$\text{Minimizar } f(x) \text{ sujeita a } H(x) = 0,$$

onde

$$H(x) = \begin{pmatrix} H_1(x) \\ \cdots \\ H_{m+q}(x) \end{pmatrix} = \begin{pmatrix} h(x) \\ g_1(x) \\ \cdots \\ g_q(x) \end{pmatrix}.$$

Pelas hipóteses desta seção, o problema auxiliar satisfaz todas as suposições da seção anterior. Mais ainda, os multiplicadores associados a x^* no problema auxiliar são os mesmos que os associados ao problema original.

Portanto, pelo Teorema 8.1, só é necessário provar que a aplicação do Algoritmo 7.1 ao problema original é equivalente à aplicação do Algoritmo 8.1 ao problema auxiliar e que a condição $\varepsilon_k \leq \eta_k \max\{\|h(x^k)\|_\infty, \|V^k\|_\infty\}$ é equivalente a $\varepsilon_k \leq \eta_k \|H(x^k)\|_\infty$.

Seja i tal que $g_i(x^*) < 0$. Pela Proposição 7.1, temos que $\mu_i^k = 0$ para k suficientemente grande.

Logo, por (7.3) e as atualizações de λ^{k+1} e μ^{k+1},

$$\|\nabla f(x^k) + \sum_{i=1}^{m} \lambda_i^{k+1} \nabla h_i(x^k) + \sum_{i=1}^{q} \mu_i^{k+1} \nabla g_i(x^k)\| \leq \varepsilon_k$$

para k suficientemente grande. Pela independência linear dos gradientes $\nabla h_i(x^*), i = 1, \ldots, m, \nabla g_j(x^*), j = 1, \ldots, q$, e a convergência de x^k, temos, como no Lema 8.1, que

$$\lim_{k\to\infty} \lambda^k = \lambda^* \text{ e } \lim_{k\to\infty} \mu^k = \mu^*.$$

Em particular, como $\mu_i^* > 0$ para todo $i \leq q$,

$$\mu_i^k > 0$$

para k suficientemente grande.

Como $\lambda^* \in (\lambda_{min}, \lambda_{max})$ e $\mu_i^* < \mu_{max}$, temos também que

$$\bar{\mu}_i^k = \mu_i^k, \ i = 1, \ldots, p$$

e

$$\bar{\lambda}_i^k = \lambda_i^k, \ i = 1, \ldots, m$$

para k suficientemente grande.

Vejamos agora que a fórmula de atualização de μ_i^{k+1} dada pelo Algoritmo 7.1 coincide com a fórmula de atualização do mesmo multiplicador, dada pelo Algoritmo 8.1, para o problema auxiliar.

Com efeito, por (7.9) e $\bar{\mu}_i^k = \mu_i^k$, temos que, para k suficientemente grande:

$$\mu_i^{k+1} = \max\{0, \mu_i^k + \rho_k g_i(x^k)\}.$$

Mas, pela complementaridade estrita e $\mu^k \to \mu^*$, $\mu_i^k > 0$ para k suficientemente grande. Portanto, para k suficientemente grande,

$$\mu_i^{k+1} = \mu_i^k + \rho_k g_i(x^k), i = 1, \ldots, q.$$

Ou seja, em termos do problema auxiliar,

$$\mu_i^{k+1} = \mu_i^k + \rho_k H_i(x^k), i = 1, \ldots, q.$$

Portanto, os multiplicadores correspondentes às restrições ativas de desigualdade se atualizam, no Algoritmo 7.1, pela mesma regra do Algoritmo 8.1 aplicada ao problema auxiliar.

Vejamos agora o significado de V_i^k. Pelo Algoritmo 7.1, temos

$$V_i^k = \max\left\{g_i(x^k), -\frac{\bar{\mu}_i^k}{\rho_k}\right\}$$

para todo $i = 1, \ldots, p$.

Agora, se $i > q$, como $g_i(x^*) < 0$, g_i é contínua e $\bar{\mu}_i^k = 0$, temos que $V_i^k = 0$ para k suficientemente grande.

Suponhamos agora que $i \leq q$. Se $g_i(x^k)$ fosse menor que $-\frac{\bar{\mu}_i^k}{\rho_k}$, pela fórmula de atualização de μ^{k+1} no Algoritmo 7.1, teriamos que $\mu_i^{k+1} = 0$. Isto é impossível para k grande devido à complementaridade estrita. Logo, para k suficientemente grande, $V_i^k = g_i(x^k)$.

Assim, para k suficientemente grande,

$$\|H(x^k)\|_\infty = \left\| \begin{pmatrix} h(x^k) \\ V^k \end{pmatrix} \right\|_\infty.$$

Isto implica que o teste para atualizar o parâmetro de penalidade é o mesmo no Algoritmo 7.1 e no Algoritmo 8.1 aplicado ao problema auxiliar. Mais ainda, a condição $\varepsilon_k \leq \eta_k \max\{\|V^k\|_\infty, \|h(x^k)\|_\infty\}$ também é equivalente à condição $\varepsilon_k \leq \eta_k \|H(x^k)\|_\infty$ imposta no Teorema 8.1.

Portanto, para k suficientemente grande, a seqüência gerada pelo Algoritmo 7.1 pode ser pensada como sendo gerada pelo Algoritmo 8.1, junto com seus multiplicadores associados e parâmetros de penalidade. **QED**

EXERCÍCIOS

1. Se diz que uma seqüência de números reais $\{r_k\}$ converge a zero com taxa R-linear pelo menos $\tau < 1$ se existe uma seqüência $\{r'_k\}$ de números positivos tal que

$$r'_{k+1} \leq \tau r'_k$$

e $|r_k| \leq r'_k$ para todo k.

Prove que, sob as hipóteses dos teoremas deste Capítulo, as seqüências $\|x^k - x^*\|$, $\|\lambda^k - \lambda^*\|$, $\|V^k\|$ e $\|\nabla L_0(x^k, \lambda^{k+1}, \mu^{k+1})\|$ convergem a zero com taxa R-linear pelo menos τ. Detecte este comportamento nas saídas de `Algencan`.

2. Considere o Algoritmo 7.1 com a seguinte modificação: o parâmetro de penalidade é multiplicado por γ em todas as iterações, independentemente do decréscimo do valor de $\|h(x^k)\|$. Percorra as provas de convergência global e comprove que todas se verificam. Agora pule para o Capítulo 8 e verifique a satisfação dos lemas que precedem o Teorema 8.1. Percorra agora a prova do Teorema e comprove que, para o Algoritmo modificado, $\|h(x^k)\|$ converge a zero superlinearmente. Usando que $\nabla f(x^k) + \nabla h(x^k)\lambda^{k+1} = 0$, deduza que o par (x^k, λ^k) converge a zero superlinearmente. Significa isto o Algoritmo de Lagrangiano Aumentado assim modificado deve ser melhor que o Algoritmo 7.1? Discuta esta possibilidade.

3. Considere o Algoritmo de Lagrangiano Aumentado 7.1. Suponha que $\{x^k\}_{k \in K}$ é uma subseqüência convergente ao ponto admissível x^* e que x^* satisfaz a Condição de Qualificação Mangasarian-Fromovitz. Prove que $\{\lambda^{k+1}, \mu^{k+1}\}$ admite alguma subseqüência convergente e que o limite de qualquer subseqüência convergente é um vetor de multiplicadores em x^*.

4. Suponha que, por sorte, você acertou com o ponto inicial e os multiplicadores corretos para a aplicação do Algoritmo 7.1. Percorra o algoritmo e descubra o que acontece com as seqüências de iterandos x^k e multiplicadores. Que acontece se você acertou com o ponto inicial mas não com os multiplicadores corretos?

5. Exiba um exemplo onde o Método de Lagrangiano Aumentado gere duas subseqüências que convergem a diferentes limites.

6. Prove as condições de segunda ordem para minimizador local do problema definido neste capítulo.

7. Estenda os resultados deste capítulo para o caso em que

$$\Omega = \{x \in I\!R^n \mid \underline{h}(x) = 0,\ \underline{g}(x) \leq 0\}$$

considerado no Capítulo 7.

Capítulo 9

Implementação e Uso de ALGENCAN

`Algencan` é um método computacional para resolver o problema PNL formulado como

$$\text{Minimizar } f(x),$$

sujeita a

$$h(x) = 0, \ g(x) \leq 0, a^{min} \leq x \leq a^{max}.$$

Como sempre, $f : \mathbb{R}^n \to \mathbb{R}, h : \mathbb{R}^n \to \mathbb{R}^m, g : \mathbb{R}^n \to \mathbb{R}^p$ e todas as funções se supõem deriváveis.

Este programa forma parte do projeto TANGO e é livremente acessável em `www.ime.usp.br/~egbirgin/tango`.

Em grandes linhas, `Algencan` é uma implementação do Algoritmo 7.1. O programa está escrito em Fortran, mas interfaces com C++, AMPL e a linguagem estatística R são fornecidas. Exaustivamente comentado, o usuário pode obter no "site" do Projeto TANGO toda a informação necessária para seu uso. O uso direto do programa em Fortran exige que o usuário forneça derivadas primeiras da função objetivo e as restrições, embora uma opção em diferenças finitas evita este desconforto, pagando o preço de considerável diminuição em eficiência. A interface AMPL, por outro lado, tem como principal virtude que o usuário não precisa incluír a programação do cálculo das derivadas.

A implementação de `Algencan` muda com bastante freqüência e é muito provável que o programa acessável na rede no momento que estas linhas são lidas não seja exatamente o descrito aqui. Todavia, algumas características essenciais parecem consolidadas e serão enfatizadas neste capítulo.

9.1 Algoritmo para o Subproblema

Para resolver o subproblema de minimizar o Lagrangiano Aumentado no Algoritmo 7.1 usamos um método para minimizar em caixas. O esquema básico deste método é o de `Gencan`. Isto é, se usa a estratégia de restrições ativas e se abandonam as faces, quando necessário, usando o gradiente espectral projetado. Entretanto, há variantes em relação ao algoritmo interno, usado dentro de cada face da caixa.

As variações disponíveis dependem das características do problema. Para problemas com poucas variáveis (digamos, menos que 100) o algoritmo interno é Newton, com globalização com regiões de confiança [4]. O mesmo algoritmo interno é usado para problemas com muitas variáveis e com uma estrutura favorável da Hessiana do Lagrangiano Aumentado embora, por enquanto, não seja totalmente claro qué significa "estrutura favorável" neste caso. Ambas opções precisam de Hessianas analíticas, o que não representa nenhuma dificuldade em caso de ser usada a interface AMPL, mas é bastante indesejável em caso de que a programação das derivadas segundas fique por conta do usuário. Se o problema tem poucas variáveis a aproximação das derivadas segundas por diferenças finitas é perfeitamente admissível, mas se o problema é grande esta alternativa é trabalhosa e precisa de análises cuidadosas da estrutura da matriz Hessiana.

No momento em que este capítulo é escrito, a versão padrão para o algoritmo usado dentro das faces é o método Newton-truncado, com quocientes incrementais para os produtos matriz-vetor e precondicionadores BFGS [7], como descrito no Capítulo 5 deste livro. Nesta versão as derivadas segundas analíticas não precisam ser fornecidas pelo usuário.

9.2 Parâmetro de Penalidade Inicial

Para bem ou para mal, a escolha de ρ_1 tem bastante influência no desempenho prático do Método de Lagrangiano Aumentado. Entretanto, uma ponderação preliminar deve ser feita: em qualquer problema prático, o usuário deve ter alguma preocupação pelo *escalamento* de suas restrições e função objetivo. Essencialmente, depois de um bom pre-processamento, o sentimento de inviabilidade de (digamos) $h_i(x) = 1$ e $g_j(x) = 1$ será o mesmo para todos os índices $i = 1, \ldots, m$ e todos os índices $j = 1, \ldots, p$. Em `Algencan` entendemos que o pre-processamento que leva a um bom escalamento do PNL é tarefa do usuário. Ou seja, `Algencan` não escala automaticamente as restrições.

Agora, a relação entre escalamento inicial e parâmetro de penalidade inicial é evidente, pois o parâmetro de penalidade inicial nada mais é do que um escalamento da função objetivo em relação ao conjunto das restrições. Se, em determinado problema, multiplicamos todas as restrições por 45^2, o resultado de aplicar o Algoritmo 7.1 seria o mesmo que se multiplicamos o parâmetro de penalidade inicial por 45. Por isso, conselhos do tipo "Coloque seu parâmetro inicial de penalidade igual a 10" não têm muito sentido, pois 10 pode ser muito grande ou muito pequeno dependendo do escalamento das restrições em relação à função objetivo. Em outras palavras, 10 não é grande nem pequeno, assim como 10^{-6} não é necessariamente pequeno e 10^6 não é necessariamente grande.

A escolha atual padrão de ρ_1 é:

$$\rho_1 = \max\left\{10^{-6}, \min\left\{10, \frac{2|f(x^0)|}{\|h(x^0)\|^2 + \|g(x^0)_+\|^2}\right\}\right\}.$$

Logo, dado que os multiplicadores iniciais padrão são nulos, quando

$$10^{-6}(\|h(x^0)\|^2 + \|g(x^0)_+\|^2) \leq 2|f(x^0)| \leq 10(\|h(x^0)\|^2 + \|g(x^0)_+\|^2),$$

o peso da função objetivo e o peso da inadmissibilidade são iguais na função objetivo do primeiro subproblema. Esta é uma maneira democrática de considerar ambos elementos para começar a otimizar. Entretanto, se $|f(x^0)|$ excede 5 vezes a medida de inadmissibilidade quadrática, optamos por $\rho_1 = 10$, sem mais justificativa que o fato de que foi a escolha que melhor funcionou em milhares de problemas da literatura. Por último, se $|f(x^0)|$ é menor que 0.5×10^{-6} vezes a medida de inadmissibilidade, optamos, também sem grande justificativa, por $\rho_1 = 10^{-6}$.

Teoricamente, a solução de um subproblema para ρ muito grande deve estar próxima da solução do problema original. Com esse critério, o aconselhável seria sempre usar valores muito grandes de ρ. Entretanto, se fizermos isso os subproblemas ficariam muito mais difíceis. A solução mais equilibrada depende sempre do problema e o usuário deve estar preparado para manipular o parâmetro de penalidade inicial de acordo com sua experiência.

9.3 Critérios de Parada

Este é um dos aspectos nos quais a Teoria de Otimização se vincula mais fortemente com a prática computacional. Implementamos algoritmos em precisão finita e dispomos de um tempo finito para resolver os problemas, entretanto nossa teoria supõe precisão infinita e fala de convergência de seqüências e subseqüências. Em que medida as decisões algoritmicas devem ser amparadas pela teoria? Não existe uma resposta teórica para esta pergunta, mas sim o resultado de muitos anos de experiência prática. Nosso ponto de vista é pragmático: na impossibilidade de testar todas as variações de implementação com chances de sucesso, o sensato é optar pelas decisões que mais consequentemente obedecem à teoria.

9.3.1 Parada nos subproblemas

Os subproblemas de `Algencan` são de minimização em caixas. As caixas definidas por $a^{min} \leq x \leq a^{max}$ são conjuntos compactos. Logo, como as funções objetivo dos subproblemas são contínuas, existe um minimizador global em todos os subproblemas. Quando o problema original não envolve limitantes naturais (a^{min} ou

a^{max}) em todas ou algumas variáveis, o usuário é convocado a postular limitantes artificiais, baseado em seu conhecimento da localização provável da solução.

Sabemos que obter a solução global dos subproblemas é, quase sempre, um objetivo demasiado ambicioso, embora desejável. Por isso, a versão implementada de `Algencan` resolve subproblemas usando `Gencan`, algoritmo para minimizar em caixas com o que, dado qualquer $\varepsilon_k > 0$ e para k suficientemente grande, um iterando para o qual a norma do gradiente projetado contínuo é menor que ε_k será fatalmente encontrado. Em outras palavras, se o número de iterações de `Gencan` é ilimitado, acabaremos encontrando um iterando z^ν tal que $a^{min} \le z^\nu \le a^{max}$ e tal que

$$\|P(z^\nu - \nabla L_{\rho_k}(z^\nu, \bar{\lambda}^k, \bar{\mu}^k)) - z^\nu\|_\infty \le \varepsilon_k.$$

(Aqui P é o operador projeção na caixa $a^{min} \le x \le a^{max}$.)

Definindo $x^k = z^\nu$, o critério teórico exigido no Algoritmo 7.1 para o iterando x^k (condições (7.3–7.6)) será satisfeito. (Ver Exercício no final deste capítulo.)

Este critério básico tem uma sólida justificativa teórica por dois motivos: fornece o tipo de iterando x^k que precisa o Algoritmo 7.1 e sua obtenção está garantida pelo algoritmo que minimize em caixas.

Na prática, entretanto, algo pode falhar. Por um lado, é possível que o tempo que `Gencan` necessite para encontrar z^ν como desejado acima seja excessivo. Se `Gencan` demora muito em determinado subproblema, parece sensato interromper a execução para recalcular os multiplicadores de Lagrange e o parâmetro de penalidade, com a expectativa de encontrar subproblemas mais fáceis. Por isso, `Algencan` estabelece um máximo padrão de 5000 iterações para a resolução de cada subproblema, máximo que, naturalmente, pode ser modificado pelo usuário, alterando o valor da variável *maxtotit* no programa principal.

Mais frequente é a possibilidade de que a condição de parada mencionada acima para o subproblema não possa ser atingida por razões de "escalamento". Isto tem a ver exclusivamente com o fato de que trabalhamos com uma representação finita do contínuo. A caixa $a^{min} \le x \le a^{max}$ tem uma quantidade finita de pontos nesta representação e, naturalmente, em um deles o valor da função L_{ρ_k} é o menor possível. Entretanto, pode ser que, nem neste ponto nem em nenhum outro, a norma do gradiente projetado contínuo seja menor que ε_k. Uma situação deste tipo é fácil de simular. Suponha que no melhor ponto da representação finita (na máquina) da caixa, tenhamos que a norma do gradiente projetado contínuo é 10^{-8} (um número com aspecto pequeno!) e que $\varepsilon_k = 10^{-4}$. Portanto, o melhor ponto na caixa da máquina satisfaz o critério de parada preconizado. Entretanto, suponha agora que a função objetivo Lagrangiano Aumentado é a anterior multiplicada por 10^5. Teremos assim que no melhor ponto a norma do gradiente projetado contínuo é 10^{-3} e, portanto, o critério de parada não é satisfeito, apesar do problema ser totalmente equivalente ao anterior. Situações como esta acontecem de fato no Lagrangiano Aumentado na prática quando o parâmetro de penalidade fica muito grande.

Quando situações como a delinhada acima acontecem, dizemos que a precisão ε_k não pode ser atingida. O comportamento de `Gencan`, nestes casos, é típico: vemos o algoritmo convergir, mas a norma do gradiente não decresce ao valor que desejávamos. `Gencan` está preparado para detectar esta eventualidade e, quando ela sucede, pára e devolve o controle para `Algencan` com o melhor ponto obtido.

Na teoria do Algoritmo 7.1 vemos que o único requisito para ε_k é que deve tender a zero. Ou seja, os subproblemas devem ser resolvidos com mais e mais precisão em cada iteração de `Algencan`. Um argumento de bom senso muito utilizado na Análise Numérica diria que, nas primeiras iterações de `Algencan`, não é sensato resolver os subproblemas com alta precisão porque, afinal de contas, a solução dos primeiros subproblemas pode estar ainda distante da solução do problema principal. Provavelmente existe uma estratégia ótima para vincular, dinamicamente, a escolha de ε_k com γ_k e τ_k mas, no momento em que estas notas são escritas, tal estratégia ainda não foi encontrada. Por isso, a versão padrão de `Algencan`, neste momento, é usar ε_k fixo e pequeno, em todas as iterações. Gostariamos de usar $\varepsilon_k = 0$ mas isto é inviável em precisão finita, pois via de regra, o critério de parada de `Gencan` jamais seria atingido. Por isso, na atual versão padrão, estabelecemos $\varepsilon_k = \varepsilon_{opt}$ para todo k. O significado de ε_{opt} será dado na seção 9.3.3.

9.3.2 Decisões emergenciais

Decidir o quê fazer quando tudo funciona exatamente dentro dos parâmetros da teoria é relativamente simples. O complicado é tomar decisões quando as coisas parecem não funcionar como previsto.

Consideremos, por exemplo, o caso em que depois do número padrão de iterações internas (5000) o critério de convergência de `Gencan` não foi atingido. Na teoria (no mundo contínuo) isto não é possível se o número de iterações permitido for suficientemente grande, pois o domínio $a^{min} \leq x \leq a^{max}$ é compacto e os teoremas de convergência de `Gencan` garantem que um ponto com o gradiente projetado contínuo tão pequeno como se queira será encontrado. Entretanto, na prática computacional, a mencionada falha é possível por dois motivos:

(a) O número de iterações não foi suficiente para atingir a precisão requerida;

(b) O critério de parada de `Gencan` jamais poderia ser atingido devido ao escalamento do problema.

De fato, não sabemos se estamos diante da situação (a) ou da situação (b). Mas temos uma gama de possíveis decisões a ser tomadas:

1. Poderiamos executar mais 5000 iterações de `Gencan`. Entretanto, se depois

deste procedimento continuássemos sem atingir o critério de convergência de `Gencan` estaríamos na mesma situação, e obrigados a tomar uma decisão emergencial. Portanto, esta não é uma verdadeira alternativa e deve ser descartada em conseqüência.

2. Depois de uma iteração problemática como a mencionada, deveriamos atualizar os multiplicadores de Lagrange ou não? Talvez haja bons argumentos para não atualizar nestes casos. Entretanto, dado que com os multiplicadores de Lagrange anteriores não fomos muito bem, nos parece sensato atualizar os multiplicadores, com a expectativa de obter resultados melhores na próxima. Um argumento adicional para *sim* atualizar os multiplicadores é a possibilidade de que nos encontemos no caso (b) mencionado acima. Neste caso, embora não tenhamos atingido o critério de convergência, estamos provavelmente muito perto da solução do subproblema e devemos proceder como se a solução do subproblema tivesse sido atingida. Logo, em `Algencan` os multiplicadores de Lagrange *são atualizados* independentemente de que o critério de convergência de `Gencan` tenha sido atingido ou não.

3. Devemos atualizar o parâmetro de penalidade depois da chamada problemática a `Gencan`? A decisão é duvidosa no caso (a), mas inequívoca no caso (b). Este caso equivale a ter atingido a precisão desejada em `Gencan`, pois pedir uma precisão exagerada veio de uma avaliação incorreta do usuário sobre o escalamento do problema. Portanto, e dado que os dois casos são indistingüíveis, a prudência nos leva a proceder como se estivéssemos sempre no caso (b). Logo, o parâmetro de penalidade é atualizado de acordo com a comparação usual das medidas de admissibilidade-complementaridade em duas iterações consecutivas. Entretanto, existe uma exceção a esta regra, que será comentada mais adiante.

Suponhamos que na iteração $k-1$ de `Algencan` a norma da admissibilidade-complementaridade seja menor que ε_{feas} e que na iteração k também, embora nesta iteração tal norma tenha aumentado em relação à anterior. Devemos multiplicar o parâmetro de penalidade por γ também neste caso? Primeiro vejamos em quê circunstâncias isto pode acontecer.

(a) No primeiro caso, o critério de convergência de `Gencan` foi satisfeito normalmente com a tolerância ε_k. Na nossa presente implementação usamos $\varepsilon_k = \varepsilon_{opt}$ para todo k, portanto neste caso x^k seria a solução devolvida por `Algencan` e não há nenhuma decisão a tomar.

(b) No segundo caso, `Gencan` não conseguiu parar com a tolerância requerida. Neste caso, julgamos que aumentar o parâmetro de penalidade somente iria a piorar o condicionamento e, portanto, as dificuldades para `Gencan`. Portanto, nos parece injusto aumentar o parâmetro de penalidade neste caso.

(c) Se não usássemos $\varepsilon_k = \varepsilon_{opt}$ a decisão poderia ser diferente, pois poderia ser que x^k não fosse a solução final devolvida por `Algencan`. Entretanto, na dúvida, decidimos proceder igual que no caso (b).

9.3.3 Parada final

A execução de `Algencan` se interrompe declarando Convergência, quando as seguintes condições são verificadas para $\varepsilon_{feas}, \varepsilon_{opt} > 0$ dados. (Na versão padrão, $\varepsilon_{feas} = \varepsilon_{opt} = 10^{-4}$.)

$$\|P(x^k - \nabla L_{\rho_k}(x^k, \bar{\lambda}^k, \bar{\mu}^k)) - x^k\|_\infty \leq \varepsilon_{opt}, \tag{9.1}$$

$$\|h(x^k)\|_\infty \leq \varepsilon_{feas}, \; g_i(x^k) \leq \varepsilon_{feas} \text{ para todo } i = 1, \ldots, p, \tag{9.2}$$

$$\mu_i^{k+1} = 0 \quad \text{para todo } i \quad \text{tal que} \quad g_i(x^k) < -\varepsilon_{feas}. \tag{9.3}$$

Devido às definições (7.7) e (7.9) de λ^{k+1} e μ^{k+1}, a condição (9.1) é equivalente a

$$\|P(x^k - \nabla L_0(x^k, \lambda^{k+1}, \mu^{k+1})) - x^k\|_\infty \leq \varepsilon_{opt}.$$

A satisfação desta condição é responsabilidade do método que resolve os subproblemas. Como dito na seção anterior, a continuidade da função objetivo L_{ρ_k} dos subproblemas e a compacidade do domínio garantem que a condição (9.1) será atingida, independentemente de qualquer outra característica do PNL. Mais ainda, se fazemos $\varepsilon_k = \varepsilon_{opt}$ para todo k, a condição é atingida por *todos* os iterandos x^k do método de Lagrangiano Aumentado.

As condições de admissibilidade (9.2) podem ser satisfeitas ou não. Por exemplo, se o problema original tem região admissível vazia (coisa que, a priori, não podemos verificar) elas jamais serão satisfeitas. Nesse caso, `Algencan` poderá parar por duas razões:

1. Durante `maxitncp` (igual a 9 por padrão) iterações consecutivas não houve progresso na admissibilidade de x^k;

2. Chegou-se a um máximo de $k =$ `maxoutit` iterações. (`maxoutit` igual a 50 no padrão.)

As condições de complementaridade (9.3) admitem uma análise mais interessante, dada pela Proposição 7.1. Suponhamos que o método está convergindo a um ponto x^*. (Mais propriamente, uma subseqüência gerada pelo método está convergindo a x^*.) Segundo a Proposição 7.1, o multiplicador μ_i^{k+1} será igual a zero para k suficientemente grande sempre que $g_i(x^*) < 0$. Poderia ser que, para infinitas iterações da subseqüência convergente, tenhamos que $g_i(x^k) < -\varepsilon_{feas}$ e $\mu_i^{k+1} > 0$? Se isto fosse possível, por continuidade, teriamos que $g_i(x^*) < 0$. Mas a Proposição 7.1 diz que, nesse caso, μ_i^{k+1} seria igual a zero para k suficientemente grande. Portanto, a condição de parada (9.3) necessariamente é satisfeita, para k suficientemente grande, independentemente de que o método convirja a um ponto admissível ou não.

Resumindo, as três condições para declarar convergência do método têm os seguintes diferentes status:

1. A condição de Lagrange (9.1) é satisfeita por todas as iterações x^k do método, a menos que o algoritmo computacional usado para minimizar em caixas falhe.

2. A condição de admissibilidade (9.2) pode ou não ser satisfeita. Quando não é satisfeita o método pára por esgotamento do número máximo de iterações.

3. A condição de complementaridade (9.3) necessariamente é satisfeita, para k suficientemente grande, exceto falha do algoritmo usado para minimizar em caixas.

Poderiamos sintetizar o anterior dizendo que a única condição crucial para decidir a parada do método é a de admissibilidade, dado que as outras duas (fora falhas computacionais) são necessariamente satisfeitas, uma em todas as iterações e outra para k suficientemente grande.

Um ponto que satisfaz as condições de parada (9.1–9.3) merece ser chamado, com toda justiça, de $\varepsilon - KKT$, com $\varepsilon = \max\{\varepsilon_{feas}, \varepsilon_{opt}\}$. Podemos dizer que o Algoritmo de Lagrangiano Aumentado, se converge a um ponto admissível, necessariamente pára em um ponto KKT aproximado. É curioso e instrutivo o fato de que isto aconteça mesmo quando o ponto limite não é KKT, e mesmo quando não existe nenhum ponto KKT. Nestes casos, algum multiplicador de Lagrange aproximado $\|\lambda^{k+1}\|$ ou $\|\mu^{k+1}\|$ tende a infinito e a Condição de Qualificação não se cumpre no limite. O usuário de `Algencan` intuirá este fato ao observar estimativas muito grandes dos multiplicadores, mas dificilmente se interesse por esta anomalia.

9.4 Usando `Algencan`

`Algencan` pode ser usado na sua forma original ou através de suas interfaces. A interface mais importante é com AMPL porque mediante ela o usuário pode se livrar do desconforto de calcular derivadas. Nesta seção falaremos exclusivamente do uso de `Algencan` na sua forma básica.

`Algencan` é um programa exaustivamente comentado. Qualquer pessoa com conhecimentos rudimentares de Fortran pode usá-lo para resolver problemas de otimização com muitas variáveis e restrições. O recomendável é que o eventual usuário imprima o programa fonte e estude com calma as modificações que deve fazer para sua aplicação.

Para usar `Algencan` na versão Fortran, o usuário deve modificar o programa fonte na forma indicada nos comentários. Os comentários formam um completo manual do usuário, portanto aqui nos limitaremos a detalhar um exemplo.

Queremos encontrar n_p pontos na bola unitária de $\mathbb{R}^{n_d}$, com a coordenada n_d de cada um entre $-1/2$ e $1/2$, de maneira que a distância mínima entre cada par deles seja a maior possível.

Primeiro vejamos como formular este problema como um PNL. Chamaremos P^1,...,P^{n_p} aos pontos de $\mathbb{R}^{n_d}$ que queremos encontrar. Assim, o problema será:

$$\text{Maximizar } \min\{\|P^i - P^j\|, i \neq j\}$$

sujeita a $\|P^k\| \leq 1, -0.5 \leq P^k_{n_d} \leq 0.5, k = 1, \ldots, n_p$.

Isto é equivalente a:

$$\text{Minimizar } \max\{-\|P^i - P^j\|, i \neq j\}$$

sujeita a $\|P^k\| \leq 1, -0.5 \leq P^k_{n_d} \leq 0.5, k = 1, \ldots, n_p$.

Acrescentando a variável auxiliar z, este problema é:

$$\text{Minimizar } z$$

sujeita a

$$-z - \|P^i - P^j\|^2 \leq 0 \;\; \forall i \neq j,$$

$$\|P^k\|^2 - 1 \leq 0 \;\; \forall k = 1, \ldots, n_p,$$

$$-0.5 \leq P^k_{n_d} \leq 0.5 \;\; \forall k = 1, \ldots, n_p.$$

Esta já é uma forma PNL padrão. O número de variáveis é $n_p n_d + 1$ e o número de restrições de desigualdade (fora as canalizações) é $n_p + n_p(n_p - 1)/2$. Cada coordenada de cada ponto P^k deve ser associada a uma variável x_i. Nossa convenção será associar z a x_n e escrever:

$$(x_1, \ldots, x_n) = (P^1_1, \ldots, P^1_{n_d}, P^2_1, \ldots, P^2_{n_d}, \ldots, P^{n_p}_1, \ldots, P^{n_p}_{n_d}, z).$$

Vejamos as modificações que precisamos fazer em `Algencan` para resolver este problema.

9.4.1 Subrotina Inip

Esta é a primeira subrotina que o usuário encontrará no pacote Algencan. É aqui que é preciso indicar o número de variáveis, de restrições, o ponto inicial e o tipo de restrições (igualdade, desigualdade, linearidade).

Uma das primeiras declarações desta subrotina, assim como das outras subrotinas que compõem o pacote é implicit none. Isto significa que o usuário deve declarar o status de todas as variáveis (inteira, real, precisão dupla, lógica).

Excluíndo os comentários originais, e substituindo esses comentários por novos textos em português, a forma que pode tomar a subrotina inip é a seguinte:

```
      subroutine inip(n,x,l,u,m,lambda,rho,equatn,linear)
      implicit none
c       m é o número de restrições, n é o número de variáveis.
      integer m, n
c       equatn é uma vetor lógico que define se cada
c       restrição é de igualdade ou não.
c       linear é uma vetor lógico que define se cada
c       restrição é linear ou não.
      logical equatn(*),linear(*)
c       l é o vetor de limitantes inferiores, u é o vetor de limitantes
c       superiores, lambda é o vetor de multiplicadores de Lagrange,
c       rho o parâmetro de penalidade e x deve ser o ponto inicial.
      double precision l(*),lambda(*),rho(*),u(*),x(*)
c       i, j, k são variáveis auxiliares
      integer i, j, k
c       nnpun é o número máximo de pontos que será usado,
c       npun é o número efetivo de pontos usado
c       neste problema (n_p)
c       e ndim é a dimensão do espaço onde
c       residem esses pontos (n_d).
c       indim é o máximo número de restrições possível.
      integer nnpun, indim, npun, ndim
      parameter (nnpun = 100)
      parameter (indim = nnpun*(nnpun-1)/2)
c       indi e indj são vetores auxiliares usados para,
c       dado o número
c       de uma restrição do tipo ``distância'',
c       estabelecer quais são
c       os dois pontos P^i e P^j envolvidos nessa restrição.
      integer indi(indim), indj(indim)
```

```
c       O common ‘‘sorvete’’ cumpre a função de comunicar às subrotinas
c       que calculam função e restrições os valores
c       dados a ndim, npun
c       e os indicadores indi, indj.
      common/sorvete/npun, ndim, indi, indj
c       O usuário introduz na tela os valores de n_d e n_p.
      write(*, *) ’ ndim, npun:’
      read(*, *) ndim, npun
      write(*, *)’ ndim=’, ndim,’ npun =’, npun
c       n é o número de variáveis do problema.
      n = ndim * npun + 1
c       O ponto inicial é escolhido como (1,2,3,...,n) neste
c       exemplo.
      do i = 1,n
      x(i) = dfloat(i)
      end do
c       Limitantes inferiores e superiores:
      do i = 1,npun
      do j = 1, ndim-1
      l((i-1)*ndim+j) = - 1.d4
      u((i-1)*ndim+j) = 1.d4
      end do
      end do
      do i = 1, npun
      l(i*ndim) = -0.5d0
      u(i*ndim) = 0.5d0
      end do
c       m é o número de restrições.
      m = npun + npun*(npun-1)/2
c       Para cada k estabelecemos quais são os valores de i e j
c       que correspondem a restrição k-ésima.  Esta informação,
c       armazenada nos vetores indi e indj, é
c       passada por common às subrotinas que calculam restrições.
      k = 1
      do i = 1, npun-1
      do j = i+1, npun
      indi(k)=i
      indj(k)=j
      k = k + 1
      end do
      end do
```

```
c       Aproximação inicial dos multiplicadores de Lagrange:
      do i = 1, m
      lambda(i) = 0.0d0
      end do
c       Parâmetro de penalidade inicial:
c       Em princípio, Algencan usa o parâmetro de penalidade
c       inicial padrão.
c       Se você faz questão de usar um parâmetro de penalidade
c       definido aqui, vá até a subroutina easyalgencan e
c       coloque rhoauto = .false.
      do i = 1, m
      rho(i) = 10.0d0
      end do
c       Como todas as restrições são de desigualdade,
c       o vetor lógico
c       equatn tem todas suas componentes falsas.
      do i = 1, m
      equatn(i) = .false.
      end do
c       Como nenhuma restrição é linear, o vetor lógico
c       linear tem todas suas componentes falsas.
      do i = 1, m
      linear(i) = .false.
      end do
      return
      end
```

9.4.2 Subrotinas `evalf` e `evalg`

As subrotinas `evalf` e `evalg` devem avaliar a função objetivo e seu gradiente no ponto x. Fazendo a mesma depuração que fizemos em `inip`, estas subrotinas podem tomar a seguinte forma para resolver o problema.

```
      subroutine evalf(n,x,f,flag)
      implicit none
      integer flag,n
      double precision f
      double precision x(n)
c       Nesta, como nas outras subrotinas fornecidas pelo usuário,
c       a variável flag deve ser igual a 0 se a avaliação transcorre
c       sem dificuldades, ou seja, sem divisões por zero,
```

```
c         logaritmos de
c         números não-positivos, etcétera.  Neste problema, dada
c         a simplicidade das funções envolvidas, nenhuma
c         dificuldade pode se manifestar e, em conseqüência,
c         flag deve ser igual a zero.
c         Nos outros casos, coloque flag = 1.
        flag = 0
        f = x(n)
        return
        end

        subroutine evalg(n,x,g,flag)
        implicit none
c         Esta subrotina deve avaliar o gradiente da função definida
c         por evalg.
c         integer n, flag
c         double precision x(n), g(n)
        flag = 0
        do i = 1 ,n - 1
        g(i) = 0.0d0
        end do
        g(n) = 1.0d0
        return
        end
```

9.4.3 Subrotinas `evalc` e `evaljac`

A subrotina `evalc` deve avaliar as restrições e `evaljac` deve avaliar os gradientes das mesmas, ou seja, a matriz Jacobiana.

Vejamos a forma destas subrotinas no nosso problema.

```
        subroutine evalc(n,x,ind,c,flag)
c         As variáveis de entrada são n, x, ind.  n é o número de
c         variáveis do problema, x o ponto no qual Algencan deseja
c         avaliar uma restrição e ind o número da restrição que
c         Algencan quer avaliar.  Assim, o usuário deve programar
c         o cálculo da restrição de índice ind em um ponto
c         genérico x.
c         A saída é o número c e o indicador flag.  O número c é
c         o valor da restrição ind no ponto x.  flag tem o mesmo
```

```
c       papel que em evalf e evalg.
c       Se tudo funciona bem, o usuário deve
c       colocar flag = 0, mas se há alguma operação proibida,
c       deve-se colocar flag = 1.
      implicit none
      integer ind, flag, n
      double precision c
      double precision x(n)
      integer i, j, k
c       As próximas cinco linhas de declarações são específicas
c      para o problema
c       de otimização que estamos exemplificando.
c       ndim é a dimensão $n_d$ e npun é o número de pontos $n_p$.
c       A informação sobre ndim e npun é passada a esta subrotina
c       através do common/sorvete/ pois tanto ndim como npun foram
c       definidos na subrotina inip.
c       Também pelo mesmo common é passada a informação indi e indj.
c       Lembremos que, dado um número de restrição k, indi(k)
c       e indj(k) dão os pontos $P^i$ e $P^j$ que estão envolvidos
c       nessa restrição.
c       Tanto indi como indj foram calculados em
c       inip.
c       nnpun é o número máximo de pontos previsto para $n_p$ em
c       problemas deste tipo.  Neste caso colocamos nnpun=100.
c       De acordo com isso, a dimensão máxima que podem ter os
c       os vetores indi e indj é indim.
      integer nnpun, indim, npun, ndim
      parameter (nnpun = 100)
      parameter (indim = nnpun*(nnpun-1)/2)
      integer indi(indim), indj(indim)
      common/sorvete/npun, ndim, indi, indj
      flag = 0
c       As npun primeiras restrições são as que dizem
c       que cada ponto
c       $P^k$ deve ter norma menor ou igual que 1.
      if(ind.le.npun) then
      c = -1.d0
      do i = 1, ndim
      c = c + x((ind-1)*ndim+i)**2
      end do
      return
```

```
      endif
c       As npun*(npun-1)/2 restrições finais se referem à distância
c       entre os pontos P^i e P^j.
      i = indi(ind-npun)
      j = indj(ind-npun)
      c = - x(n)
      do k = 1, ndim
      c = c - (x((i-1)*ndim+k) - x((j-1)*ndim+k))**2
      end do
      return
      end

      subroutine evaljac(n,x,ind,indjac,valjac,nnzjac,flag)
c       Os significados das variáveis de entrada n, x, ind e da
c       variável de saída flag são os mesmos que em evalc.
c       Nesta subrotina, dado o ponto x fornecido por Algencan,
c       o usuário deve programar o gradiente da restrição ind.
c       Para isso, o usuário fornecerá as seguintes informações:
c       O número nnzjac deve dizer quantos elementos diferentes
c       de zero pode ter esse gradiente, ou, em outras palavras,
c       quantas variáveis aparecem na definição da restrição ind.
c       O vetor inteiro indjac, com nnzjac posições, diz quais
c       são as variáveis envolvidas na restrição ind.
c       Finalmente, se j = indjac(k), valjac(k) deve ser
c       a derivada
c       da restrição ind em relação a variável j.
      implicit none
      integer flag,ind,n,nnzjac,npun,ndim,nnpun,indim,i,j,k
      integer indjac(n)
      double precision x(n),valjac(n)
c       nnpun, npun, indim, indi, indj e o common/sorvete/ têm
c       o mesmo significado que em evalc.
      parameter (nnpun = 100)
      parameter (indim = nnpun*(nnpun-1)/2)
      integer indi(indim), indj(indim)
      common/sorvete/npun, ndim, indi, indj
      flag = 0
      if(ind.le.npun) then
c       As primeiras npun restrições são as que dizem que
c       todos os pontos P^i estão na bola unitária.
c       Para programar as derivadas é útil lembrar as restrições
```

```
c       nos comentários.
c       c = -1.d0
c       do i = 1, ndim
c       c = c + x((ind-1)*ndim+i)**2
c       end do
      nnzjac = ndim
      do i = 1, ndim
      indjac(i) = (ind-1)*ndim+i
      valjac(i) = 2.d0 * x((ind-1)*ndim+i)
      end do
      return
      endif
c       As restrições npun+1,...,m são as que se referem à
c       distância entre P^i e P^j.
      i = indi(ind-npun)
      j = indj(ind-npun)
c       c = - x(n)
c       do k = 1, ndim
c       c = c - (x((i-1)*ndim+k) - x((j-1)*ndim+k))**2
c       end do
      nnzjac = 2* ndim+1
      do k = 1, ndim
      indjac(k) = (i-1)*ndim+k
      indjac(ndim+k) = (j-1)*ndim+k
      valjac(k) = - 2.d0 * (x((i-1)*ndim+k) - x((j-1)*ndim+k))
      valjac(ndim+k) = - 2.d0 * (x((j-1)*ndim+k) - x((i-1)*ndim+k))
      end do
      indjac(2*ndim+1) = n
      valjac(2*ndim+1) = -1.d0
      return
      end
```

9.4.4 Programa Principal, subrotina `param` e subrotina `endp`

Muitas vezes, para resolver um problema PNL, o usuário precisará usar `Algencan` utilizando diferentes pontos iniciais. Uma estratégia simples, muito comum, consiste em partir de pontos iniciais gerados aleatoriamente, provavelmente de maneira uniforme e dentro da caixa do problema. Para fazer isso, será necessário fazer modificações no "Programa Principal" do pacote `Algencan`. Para ter uma idéia de como isto pode ser feito, vejamos aqui a estrutura padrão deste programa principal:

```
      implicit none
      integer mmax,nmax
c       nmax e mmax são o número máximo de variáveis e o número
c       máximo de restrições, respectivamente, permitido por
c       Algencan.  Estes números podem ser aumentados modificando
c       as declarações parameter que seguem.
      parameter ( mmax = 500000 )
      parameter ( nmax = 500000 )
c       As próximas linhas são declarações que não devem ser
c       mudadas pelo usuário.
      logical checkder
      character * 6 precond
      integer gtype,hptype,inform,iprint,m,maxoutit,maxtotfc,maxtotit,n,
     + ncomp,outiter,totcgcnt,totfcnt,totgcnt,totiter
      double precision epsfeas,epsopt,f,nalpsupn,snorm
      real time
      integer wi1(nmax)
      logical equatn(mmax),linear(mmax)
      double precision l(nmax),lambda(mmax),rho(mmax),u(nmax),wd1(mmax),
     + wd2(mmax),wd3(nmax),wd4(mmax),wd5(mmax),wd6(nmax),
     + wd7(nmax),wd8(nmax),wd9(nmax),wd10(nmax),wd11(nmax),
     + wd12(nmax),wd13(nmax),wd14(nmax),wd15(nmax),wd16(nmax),
     + wd17(nmax),wd18(nmax),wd19(nmax),x(nmax)
      external solver
c       A primeira tarefa do programa principal é chamar à subrotina
c       inip, onde o usuário deve ter inserido dados do problema
c       como número de variáveis e restrições, limitantes
c       inferiores e superiores, tipo de restrições e ponto inicial.
      call inip(n,x,l,u,m,lambda,rho,equatn,linear)
c       Depois de chamar a inip, o programa principal chama à
c       subrotina param, cuja função é estabelecer os
c       parâmetros com os quais Algencan tratará de resolver
c       o problema.
      call param(gtype,hptype,precond,checkder,epsfeas,epsopt,maxoutit,
     +maxtotit,maxtotfc,iprint,ncomp)
c       Em continuação, o programa principal chama à subrotina
c       solver, que, de fato, é o método de otimização.
      call solver(n,x,l,u,m,lambda,rho,equatn,linear,gtype,hptype,
     +precond,checkder,epsfeas,epsopt,maxoutit,maxtotit,maxtotfc,iprint,
     +ncomp,f,snorm,nalpsupn,outiter,totiter,totfcnt,totgcnt,totcgcnt,
     +time,inform,wi1,wd1,wd2,wd3,wd4,wd5,wd6,wd7,wd8,wd9,wd10,wd11,
```

```
      +wd12,wd13,wd14,wd15,wd16,wd17,wd18,wd19)
c       Finalmente, depois de obtido o resultado, o programa
c       principal chama à subrotina endp, que será programada pelo
c       usuário de maneira que os resultados finais lhe sejam
c       apresentados da maneira mais conveniente para seus
c       próprios objetivos.
      call endp(n,x,l,u,m,lambda,rho,equatn,linear)
      stop
      end
```

A subrotina `param` está exaustivamente comentada no pacote `Algencan`. Usuários experientes podem obter melhores desempenhos que os padrão para seus problemas manipulando de maneira inteligente os parâmetros de execução do método. Aqui comentaremos apenas algumas das modificações mais úteis.

```
      subroutine param(gtype,hptype,precond,checkder,epsfeas,epsopt,
      +maxoutit,maxtotit,maxtotfc,iprint,ncomp)
      implicit none
      logical checkder
      character * 6 precond
      integer gtype,hptype,iprint,maxoutit,maxtotfc,maxtotit,ncomp
      double precision epsfeas,epsopt
c       gtype é um dos parâmetros que o usuário será mais tentado
c       a modificar.  Se gtype = 1, Algencan calcula as derivadas da
c      da função e as restrições usando diferenças finitas.
c       Logo, se se substitui ‘‘gtype=0’’ por ‘‘gtype=1’’,
c       o usuário não precisará programar o gradiente da função
c       objetivo nem o Jacobiano das restrições.  Essa opção faz
c       diminuír bastante a eficiência de Algencan, mas muitas vezes
c       é útil.
      gtype = 0
c       A variável lógica checkder serve para checar numéricamente
c       as derivadas programadas analíticamente en evalg e evaljac.
c       Se checkder = .true., Algencan faz essa verificação em um
c       ponto aleatório e devolve os resultados na tela ao usuário
c       de maneira que este possa conferir seus cálculos analíticos.
      checkder = .false.
c       epsfeas é o parâmetro ε_feas, usado para
c       decretar se o ponto obtido é admissível ou não.
c       epsopt é o parâmetro ε_opt, a precisão usada
c       para otimalidade.
      epsfeas = 1.0d-04
```

```
      epsopt = 1.0d-04
c       maxoutit é o número máximo de iterações permitido
c       para Algencan.
c       maxtotit é o número total de iterações permitido para
c       o método que resolve os subproblemas (Gencan)
c       maxtotfc é o número máximo de avaliações de função
c       permitido na soma das chamadas a Gencan.
      maxoutit = 50
      maxtotit = 1000000
      maxtotfc = 5 * maxtotit
c       iprint serve para controlar a impressão na tela.
c       Quanto maior
c       for, maiores são os detalhes da saída que Algencan imprime.
      iprint = 2
c       Para o significado dos seguintes parâmetros,
c       é melhor ver
c       os comentários desta subrotina em Algencan.
      ncomp = 5
      precond = 'QNCGNA'
      hptype = 6
      return
      end
```

Depois de terminada a tarefa de `Algencan`, o usuário pode desejar fazer algum cálculo adicional, provavelmente com o fim de facilitar a leitura da saída ou usar os resultados para outros objetivos. Para isso, o pacote inclui a subrotina `endp`, que em sua versão padrão não faz nada. Vejamos aqui as modificações razoáveis que o usuário pode fazer para obter uma apreciação melhor dos resultados. O que `endp` fará, neste caso, é calcular a distância mínima entre os pontos obtidos e imprimir todos os pontos na tela em uma ordem razoável.

```
      subroutine endp(n,x,l,u,m,lambda,rho,equatn,linear)
      implicit none
      integer m,n, i, npun, ndim, j, k, nnpun, indim
      parameter (nnpun = 100)
      parameter (indim = nnpun*(nnpun-1)/2)
      integer indi(indim), indj(indim)
      common/sorvete/npun, ndim, indi, indj
      logical equatn(m),linear(m)
      double precision l(n),lambda(m),rho(m),u(n),x(n),z,zmin
      write(*, *)' Best set of points found:'
```

```
do i = 1, npun
write(*, *)i, (x((i-1)*ndim+j), j=1,ndim)
end do
zmin = 1.d30
do i = 1, npun-1
do j = i+1, npun
z = 0.d0
do k = 1, ndim
z = z + (x((i-1)*ndim+k)-x((j-1)*ndim+k))**2
end do
z = dsqrt(z)
if(z. le. zmin) zmin = z
end do
end do
write(*, *)' Minimal distance:', zmin
return
end
```

9.4.5 Resultado de `Algencan`

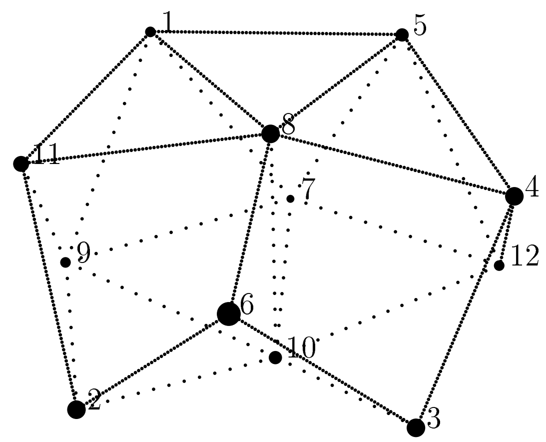
Rodamos o problema exemplo com `npun =12, ndim=3`. A parte final da impressão produzida por `Algencan` na tela foi a seguinte:

```
ALGENCAN outer iteration:  16
Current point (first 5 components):
-8.0449E-01 3.2065E-01 5.0000E-01 -2.5296E-01 -8.7202E-01
Updated Lagrange multipliers (first 5 components):
5.6186E-02 9.7019E-02 7.8359E-02 9.5448E-02 7.9739E-02
Updated penalty parameters (first 5 components):
1.9129E+02 1.9129E+02 1.9129E+02 1.9129E+02 1.9129E+02
Objective function value = -8.8532E-01
Maximal violation of feasibility-complementarity = 1.8306E-05
Sup-norm of the projected gradient of the Lagrangian = 9.8352E-06
Up-to-now number of inner iterations = 824
Up-to-now number of Augmented Lagrangian function evaluations = 2501
Up-to-now number of Augmented Lagrangian gradient evaluations = 931
Flag of ALGENCAN = 0 (Convergence with Sup-norm of the Lagrangian projected
gradient smaller than 1.0000E-04, maximal infeasibility smaller than
1.0000E-04, and largest complementarity violation smaller
than 1.0000E-04)
```

A impressão gerada pela subrotina `endp` foi a seguinte:

```
Best set of points found:
1 -0.804490207 0.320645932 0.5
2 -0.252960237 -0.872019826 -0.419039034
3 0.879020515 -0.273464861 -0.390562788
4 0.883830201 0.331500992 0.330073625
5 -0.0371920266 0.865225355 0.5
6 0.453823505 -0.86788081 0.20205891
7 -0.511111833 0.801516183 -0.310353757
8 0.0507746088 -0.0715736089 0.5
9 -0.932058875 -0.0385117279 -0.36025892
10 -0.00424318626 0.0318093299 -0.5
11 -0.711344807 -0.604817427 0.358012737
12 0.431493967 0.845105674 -0.3156249
Minimal distance:  0.940910191
```

Desenhando estes pontos no espaço, obtemos a seguinte figura:



EXERCÍCIOS

1. Uma situação onde a influência do parâmetro de penalidade inicial aparece de maneira dramática é quando as restrições vêm da discretização de um problema de contorno de segunda ordem. Ao discretizar uma derivada segunda o passo de discretização (elevado ao quadrado) aparece no denominador, mas a equação poderia ser multiplicada por esse passo quadrado, o que muda totalmente o escalamento. Analise este caso na prática e reflita sobre o mesmo.

2. Em um exercício do Capítulo 8.1 foi proposta uma modificação do Algoritmo 7.1 e foi observado que, com esta modificação, o algoritmo resultava superlinear. Pediu-se para discutir se isto significaria que a modificação realmente melhorava o algoritmo. Desenhe experimentos com `Algencan` para analisar na prática esta possibilidade. Você pode forçar o crescimento de ρ_k em todas as iterações modificando o parâmetro que joga o papel de τ em `Algencan`.

3. Prove que o critério de parada nos subproblemas de `Algencan` coincide com o critério de parada para os subproblemas colocado no Algoritmo 7.1 (condições (7.3–7.6). Para isto, defina $\underline{m} = 0, \underline{p} = 2n$, $\underline{g}_i(x) = x_i - a_i^{max}, i = 1, \ldots, n$, $\underline{g}_{n+i}(x) = a_i^{min} - x_i, i = 1, \ldots, n$ e encare as contas (um pouco tediosas, certamente). Observe que, em `Gencan`, todos os iterandos pertencem à caixa $a^{min} \leq x \leq a^{max}$, portanto $\underline{g}(x^k) \leq 0$.

Referências Bibliográficas

[1] R. Andreani, E. G. Birgin, J. M. Martínez and M. L. Schuverdt, *On Augmented Lagrangian Methods with Arbitrary Lower-Level Constraints*, Technical Report MCDO-050913 (see `www.ime.usp.br/~martinez`), Department de Applied Mathematics, UNICAMP, Brazil, 2005.

[2] R. Andreani, E. G. Birgin, J. M. Martínez, M. L. Schuverdt. Augmented Lagrangian methods under the Constant Positive Linear Dependence constraint qualification. *Mathematical Programming*, por aparecer.

[3] R. Andreani, J. M. Martínez, M. L. Schuverdt. On the relation between Constant Positive Linear Dependence Condition and Quasinormality Constraint Qualification. *Journal of Optimization Theory and Applications* 125, pp. 473-485 (2005).

[4] M. Andretta, E. G. Birgin, J. M. Martínez. Practical active-set Euclidian trust-region method with spectral projected gradients for bound-constrained minimization. *Optimization* 54, pp. 305-325 (2005).

[5] D. P. Bertsekas. *Nonlinear Programming*, Athena Scientific, Belmont, Massachusetts, 1999.

[6] E. G. Birgin, J. M. Martínez. Large-scale active-set box-constrained optimization method with spectral projected gradients. *Computational Optimization and Applications* 23, pp. 101-125 (2002).

[7] E. G. Birgin, J. M. Martínez. Structured minimal-memory inexact quasi-Newton method and secant preconditioners for Augmented Lagrangian Optimization. *Computational Optimization and Applications*, por aparecer.

[8] E. G. Birgin, J. M. Martínez, M. Raydan. Inexact Spectral Projected Gradient methods on convex sets. *IMA Journal on Numerical Analysis* 23, pp. 539-559 (2003).

[9] E. G. Birgin, J. M. Martínez, M. Raydan. Algorithm 813 SPG: Software for convex-constrained optimization. *ACM Transactions on Mathematical Software* 27, pp. 340-349 (2001).

[10] E. G. Birgin, J. M. Martínez, M. Raydan. Nonmonotone spectral projected gradient methods on convex sets. *SIAM Journal on Optimization* 10, pp. 1196-1211 (2000).

[11] A. R. Conn, N. I. M. Gould, Ph. L. Toint. *Trust Region Methods*, MPS/SIAM Series on Optimization, SIAM, Philadelphia, 2000.

[12] A. V. Fiacco, G. P. McCormick. *Nonlinear Programming: Sequential Unconstrained Minimization Techniques*, John Wiley and Sons, New York, 1968.

[13] R. Fletcher. *Practical Methods of Optimization*, Academic Press, London, 1987.

[14] R. Fourer, D. M. Gay, B. W. Kernighan. *AMPL: A Modeling Language for Mathematical Programming*, Duxbury Press / Brooks/Cole Publishing Company, 2002.

[15] A. Friedlander, J. M. Martínez, B. Molina, M. Raydan. Gradient methods with retards and generalizations. *SIAM Journal on Numerical Analysis* 36, pp. 275-289 (1998).

[16] A. Friedlander, J. M. Martínez, S. A. Santos. A new trust region algorithm for bound constrained minimization. *Applied Mathematics and Optimization* 30, pp. 235-266 (1994).

[17] M. R. Hestenes. Multiplier and Gradient Methods. *Journal of Optimization Theory and Applications* 4, pp. 303-320 (1969).

[18] A. N. Iusem. Augmented Lagrangian and Proximal Point Methods for Convex Optimization, *Investigación Operativa* 8, pp. 11-49 (1999).

[19] M. J. D. Powell. A method for nonlinear constraints in minimization problems. In *Optimization*, R. Fletcher (editor), Academic Press, New York, pp. 283-298 (1969).

[20] M. Raydan. The Barzilai and Borwein gradient method for the large scale unconstrained minimization problem, *SIAM Journal on Optimization*, 7, pp. 26–33 (1997).

[21] R. T. Rockafellar. Augmented Lagrange multiplier functions and duality in nonconvex programming. *SIAM Journal on Control and Optimization* 12, pp. 268-285 (1974).